

VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA
EKONOMICKÁ FAKULTA

KATEDRA SYSTÉMOVÉHO INŽENÝRSTVÍ

Návrh simulátoru založeného na umělých neuronových sítích pro
identifikaci vstupně výstupních funkcí systémů

Proposal of the Simulator for the Identification of Input Output
Functions of the Systems Based on Artificial Neural Networks

Student: Bc. Martin Jurášek

Vedoucí diplomové práce: prof. Ing. Dušan Marček, CSc.

Ostrava 2015

VŠB - Technická univerzita Ostrava
Ekonomická fakulta
Katedra systémového inženýrství

Zadání diplomové práce

Student: **Bc. Martin Jurášek**

Studijní program: N6209 Systémové inženýrství a informatika

Studijní obor: 6209T025 Systémové inženýrství a informatika

Téma: Návrh simulátoru založeného na umělých neuronových sítích pro identifikaci vstupně výstupních funkcí systémů
Proposal of the Simulator for the Identification of Input Output Functions of the Systems Based on Artificial Neural Networks

Zásady pro vypracování:

1. Úvod
2. Teoretické východiska modelování časových řad a principu umělých neuronových sítí
3. Návrh a vytvoření simulátoru pro simulování činností neuronových sítí
4. Ověření a porovnání predikční přesnosti simulátoru se statistickými modely
5. Závěr

Seznam použité literatury

Seznam zkratk

Prohlášení o využití výsledků diplomové práce

Seznam příloh

Přílohy

Seznam doporučené odborné literatury:

KECMAN, Vojislav. *Learning and soft computing*. Massachusetts: MIT Press, 2001. 541 p. ISBN 02-621-1255-8.

MARČEK, Dušan a Milan MARČEK. *Neurónové siete a ich aplikácie*. Žilina: EDIS, 2006. 223 s. ISBN 80-8070-497-X.

MARČEK, D., M. MARČEK a L. PANČÍKOVÁ. *Ekonometria a soft computing*. Žilina: EDIS, 2008. 271 s. ISBN 978-80-8070-7460.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **prof. Ing. Dušan Marček, CSc.**

Datum zadání: 21.11.2014

Datum odevzdání: 25.04.2015



doc. Ing. Jana Hančlová, CSc.
vedoucí katedry

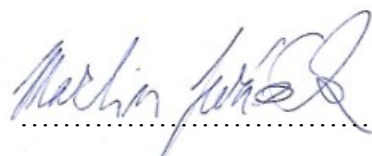




prof. Dr. Ing. Dana Dluhošová
děkanka fakulty

Místopřísežné prohlášení o samostatném vypracování diplomové práce
„Prohlašuji, že jsem celou práci, včetně všech příloh, vypracoval samostatně“.

Zároveň bych chtěl na tomto místě poděkovat svému vedoucímu diplomové práce, prof. Ing. Dušanu Marčěkovi, CSc., za odborné vedení, cenné rady a připomínky, které mi pomohly k vypracování této diplomové práce.

A handwritten signature in blue ink, appearing to read 'Martin Jurášek', written over a horizontal dotted line.

Martin Jurášek

Datum odevzdání diplomové práce: 25. dubna 2015

Obsah

1 Úvod.....	5
2 Teoretické východiska modelování časových řad a principu umělých neuronových sítí.....	7
2.1 Ekonometrie.....	7
2.1.1 Časové řady a vysokofrekvenční data	7
2.1.2 Analýza časové řady.....	8
2.1.3 Popisné charakteristiky časových řad	8
2.1.4 Stacionarita časových řad.....	9
2.1.5 Dekompozice časových řad.....	9
2.1.6 Etapy ekonometrického modelování	10
2.1.7 Box-Jenkinsova analýza dat, ARIMA procesy	13
2.2 Neuronové sítě.....	20
2.2.1 Zpracování informací neuronovou buňkou	20
2.2.2 Vícevrstvé neuronové sítě	22
2.2.3 Neuronová síť Perceptron	24
2.2.4 RBF sítě.....	27
2.2.5 Rozdělení vstupního souboru	30
3 Návrh a vytvoření simulátoru pro simulování činností neuronových sítí	32
3.1 Načítání dat.....	32
3.2 Nástroj simulace neuronové sítě typu perceptron.....	33
3.3 Nástroj simulace neuronové sítě typu RBF	34
3.4 Výstup simulátoru pro chod neuronových sítí.....	36
4 Ověření a porovnání predikční přesnosti simulátoru se statistickými modely.....	37

4.1	Akcie Ford Motor Co.	38
4.1.1	Modelování ARIMA	39
4.1.2	Modelování časové řady pomocí neuronových sítí.....	42
4.1.3	Konečné srovnání pro časovou řadu vývoje ceny akcií Ford.....	48
4.2	Akcie Pfizer Inc.	49
4.2.1	Modelování ARIMA	51
4.2.2	Modelování časové řady pomocí neuronových sítí.....	53
4.2.3	Konečné srovnání pro časovou řadu vývoje ceny akcií Pfizer.....	58
4.3	Akcie Cisco Systems Inc.	58
4.3.1	Modelování ARIMA	60
4.3.2	Modelování časové řady pomocí neuronových sítí.....	62
4.3.3	Konečné srovnání pro časovou řadu akcií Cisco Systems	68
4.4	Celkové hodnocení	68
5	Závěr	70
	Seznam použité literatury.....	72
	Seznam zkratk.....	74
	Prohlášení o využití výsledků diplomové práce	
	Seznam příloh	
	Přílohy	

1 Úvod

Dnešní svět informatiky ve velké míře ovlivňuje obor umělé inteligence. Do tohoto oboru pak spadají také neuronové sítě. Obor umělé inteligence a zejména neuronových sítí je využitelný v mnoha oblastech lidského života. Ekonomické odvětví v dnešní době zároveň do jisté míry stojí na kvantitativních ukazatelích vývoje různých veličin. Zejména pak mnoha druhů časových řad, které se snaží lidé různými způsoby pochopit a tyto časové řady následně vývojově predikovat. Jaká by tedy byla možnost využít obor umělé inteligence, tedy konkrétně neuronových sítí, při predikci časových řad ekonomických problémů? Na tuto otázku se bude snažit tato diplomová práce nalézt odpověď.

Cílem této diplomové práce tedy je navržení a vytvoření simulátoru chodu neuronových sítí pro vybrané architektury neuronových sítí a jeho využití při následné statické predikci ekonomických časových řad, v porovnání se statickou predikcí vykonanou současnými vybranými statistickými modely.

Tato diplomová práce je vyjma první kapitoly Úvodu a páté kapitoly Závěru rozdělena do tří hlavních kapitol. V pořadí druhá kapitola popisuje teoreticko-metodologické východiska práce, které jsou nezbytné pro analýzu časových řad pomocí statistických modelů a jejich následné využití při statické predikci časových řad. A také jako teoretický základ pro pochopení architektur a postupu výpočtu chodu neuronových sítí, pro možnost vytvoření simulátoru jejich chodu. V této části jsou tak popsány a rozděleny časové řady a jejich zákonitosti, dále vysvětleny a určeny etapy ekonometrického modelování a položeny základy pro Box-Jenkinsovu metodologii a ARIMA modely. Dále pak nastíněna historie vývoje neuronových sítí, principu jejich vzniku a vysvětlení jednotlivých architektur a principu jejich fungování.

V další, v pořadí třetí kapitole je popsán návrh a vytvoření simulátoru chodu neuronových sítí. Tato kapitola je rozdělena na problematiku načítání vstupních dat a dále principům fungování nástroje simulace neuronové sítě typu RBF a neuronové sítě typu Perceptron. Vysvětlen je zde také výstup simulátoru, který poté slouží pro analýzu v další kapitole, kde tento simulátor poskytuje výstupy pro srovnávací analýzu.

Nadcházející čtvrtá kapitola je věnována ověření a porovnání predikční přesnosti simulátoru neuronových sítí s vybranými statistickými modely. Kapitola je rozdělena na tři podkapitoly, věnujícím se analýzám vybraných ekonomických časových řad vývoje cen akcií, pomocí simulátoru neuronových sítí a statistických modelů. Samozřejmostí je využití nástrojů v predikci těchto časových řad a jejich následné porovnání. Na konec této kapitoly přichází

celkové hodnocení a porovnání jednotlivých architektur neuronových sítí se statistickými modely, včetně komentáře problémů, které s sebou nese nastavování parametrů jednotlivých architektur neuronových sítí.

2 Teoretické východiska modelování časových řad a principu umělých neuronových sítí

V následující části, zabývající se teoretickými východisky modelování časových řad a principu umělých neuronových sítí bude podrobně vysvětlen a popsán mechanismus ekonometrického modelování časových řad a neuronových sítí. Zejména pak těch mechanismů, které budou sloužit pro aplikované příklady v dalších částech práce.

2.1 Ekonometrie

2.1.1 Časové řady a vysokofrekvenční data

Časová řada lze definovat jako posloupnost hodnot daných ukazatelů, jež jsou měřeny ve známých intervalech času.

Časové intervaly pak mohou být pravidelné, kvazi-pravidelné nebo nepravidelné. Samotný čas pak v časové řadě tvoří nezávislou proměnnou (Hančlová a Tvrdý, 2003).

Řadit časové řady můžeme také podle charakteru ukazatele:

- okamžikové – hodnota ukazatele je dána k určitému okamžiku,
- intervalové – závisí na délce časové intervalu, ve kterém je ukazatel sledován.

Z hlediska druhů ukazatelů můžeme časové řady rozlišit následujícím způsobem:

- absolutní ukazatele – očištěné hodnoty,
- odvozené ukazatele – součty, poměry.

Samotné časové řady se vyskytují v mnoha odvětvích. Jako příklady lze uvést následující:

Ekonomie:

- vývoj kurzu měny,
- vývoj ceny akcií.

Doprava:

- počet autonehod,
- počet registrovaných automobilů.

Počasí:

- vývoj množství srážek,
- sledování teplot.

2.1.2 Analýza časové řady

Důvody pro analýzu časových řad jsou následující:

- porozumění jednotlivých procesů, které ovlivňují vývoj časové řady,
- možnost vytvoření modelu, který bude schopen predikovat hodnoty časové řady.

Grafickou reprezentaci časových řad tvoří nejčastěji spojnicové grafy.

2.1.3 Popisné charakteristiky časových řad

Pro práci s časovými řadami je důležité zjistit následující hodnoty (Hančlová a Tvrđý, 2003).

Charakteristiky polohy:

Prostý aritmetický průměr -

$$\bar{y} = \frac{\sum_{t=1}^n y_t}{n} \quad (2.1)$$

Vážený aritmetický průměr -

$$\bar{y} = \frac{\sum_{t=1}^n v_t y_t}{\sum_{t=1}^n v_t}, \text{ kde } v_t \text{ je váha ukazatele } y_t \text{ v čase } t; \quad (2.2)$$

Charakteristiky variability:

Rozptyl

$$s_y^2 = \frac{1}{n-1} \sum_{t=1}^n (y_t - \bar{y})^2 \quad (2.3)$$

$$s_y = \sqrt{s_y^2} \quad (2.4)$$

2.1.4 Stacionarita časových řad

Pokud časová řada podléhá v jejím průběhu změnám v průměru nebo variabilitě, označujeme tuto řadu za nestacionární. Stacionární časová řada tyto změny nevykazuje. U nestacionární časové řady tedy můžeme odlišit jednotlivé úseky časové řady. Tato řada má například signifikantně rozdílné průměry jejich hodnot na začátku a konci této řady.

Vykazování stacionární vlastnosti časové řady je důležitým předpokladem pro některé druhy analýz. Stacionaritu časové řady je tedy nutno testovat a v případě nutnosti ji transformovat tak, aby došlo k odstranění nestacionarity. (Hančlová a Tvrď, 2003)

2.1.5 Dekompozice časových řad

V rámci analýzy časové řady vycházíme z předpokladu, že časová řada může obsahovat následující složky:

- a) Trend – složka, která zachytává dlouhodobé změny v průměrném vývoji časové řady (tedy například dlouhodobý růst nebo pokles),
- b) Sezónní složku – tato složka popisuje pravidelné a opakující se změny časové řady. Tyto změny mají určitou časovou periodu (tady například týden, měsíc). Rozbor této složky časové řady pak přináší lepší poznání zkoumaného jevu zachyceného v časové řadě. Časová řada, která je očištěná od sezónní složky umožňuje lépe studovat dlouhodobé tendence časové řady,
- c) Cyklickou složku – taktéž periodická složka časové řady, avšak perioda této složky neodpovídá kalendářním jednotkám tak, jako u sezónní složky. Pro určení této složky se používá například metoda spektrální analýzy. Zjednodušeně se dá o cyklické složce mluvit jako o kolísání hodnot časové řady okolo jejího trendu. Při tomto kolísání se pak střídají jednotlivé fáze poklesu a růstu hodnot. Nejviditelnějším příkladem cyklické složky v praxi je ekonomický cyklus, tedy růst a pokles ekonomické aktivity. Samotný

výskyt této cyklické složky pak nemusí mít nutně původ v ekonomických oblastech, ale mohou ji vytvářet také mimoekonomické faktory,

- d) Náhodnou složku – nebo také reziduální složka nebo bílý šum. Je to složka, která zůstává přítomna v časové řadě i po odstranění trendové, sezónní i cyklické složky. Je tvořena náhodným kolísáním hodnot v průběhu dané časové řady, kdy tyto hodnoty nemají rozpoznatelný systematický charakter. Lze také říci, že se jedná o nekorelované náhodné hodnoty, které vykazují nulovou střední hodnotu.

Cílem dekompozice časové řady je pak identifikovat pravidelné chování časové řady, než původní nerozložené řady. (Hančlová a Tvrđý, 2003)

2.1.6 Etapy ekonometrického modelování

Základní ekonometrický proces se skládá z následujících čtyř etap (Marček, a další, 2008):

- a) Návrh modelu
- b) Kvantifikace modelu
- c) Verifikace modelu
- d) Aplikace modelu

Základní popis jednotlivých etap ekonometrického procesu vypadá následovně.

2.1.6.1 Návrh modelu

V etapě návrhu ekonometrického modelu jde o to zkonstruovat takový model, který bude co nejlépe reprezentovat a popisovat vztahy dané časové řady. V rámci návrhu ekonometrického modelu je potřeba specifikovat jednotlivé proměnné. Proměnné veličiny ekonometrického modelu se rozdělují na veličiny závislé a nezávislé, nebo také vysvětlované a vysvětlující. Také je v této etapě potřeba navrhnout rovnice modelu, které se získají na základě dané analýzy časové řady. (Marček, a další, 2008)

V rámci výběru typu postupu pro analýzu časové řady je nutno zohlednit několik faktorů, patří mezi ně:

- softwarové a hardwarové vybavení,
- dostupný počet pozorování dané časové řady,
- účel analýzy,
- typ časové řady.

Základní postupy pro analýzu časových řad jsou následující:

- lineární dynamické modely,
- dekompozice časové řady,
- spektrální analýza časové řady,
- Box-Jenkinsova analýza dat.

Stručný popis jednotlivých postupů pro analýzu časové řady je následující (Marček, a další, 2008):

Lineární dynamické modely - jsou hlavně příčinné modely, ve kterých je závislá proměnná vysvětlována pomocí svých zpožděných hodnot a pomocí dalších faktorů. Rozdílem v této metodě a v metodě Box-Jenkinse je právě v tom, že v této metodě jsou právě použity, kromě zpožděných hodnot, další časové řady.

Dekompozice časové řady - využívá rozložení časové řady na její jednotlivé složky

Spektrální analýza časové řady - má od předchozích metod odlišný přístup. Tento přístup spočívá v tom, že se daná časová řada považuje za směs sinusových a kosinusových křivek o různých frekvencích. Pomocí statistických nástrojů se pak získává pohled na zastoupení jednotlivých prvků v časové řadě.

Box-Jenkinsova analýza dat – tato metoda je použita pro analýzu časových řad v této diplomové práci a je podrobně vysvětlena v kapitole 2.1.7

2.1.6.2 Kvantifikace modelu

Kvantifikací ekonometrického modelu se rozumí vyčíslení hodnot parametrů modelu prostřednictvím odhadových statistických nebo ekonometrických metod. (Marček, a další, 2008)

Pro vykonání kvantifikace je nutné mít k dispozici pozorované hodnoty těch proměnných veličin, které byly specifikovány v rámci etapy návrhu modelu. Současně je potřeba zvolit kterou odhadovou metodu pro odhad daných parametrů použít. Výběr této metody je závislý na předpokladu o charakteristikách náhodné složky modelu. Rozhodnutí o využití dané metody je také závislé od typu dat, počtu jejich pozorování, kvalitě dat apod.

Do rámce kvantifikace modelu patří také určení teoretických hodnot vysvětlovaných veličin. Stanovení teoretických hodnot vysvětlovaných veličin, na základě odhadnutých parametrů a v závislosti od hodnot vysvětlujících veličin se nazývá řešení modelu. (Marček, a další, 2008)

2.1.6.3 Verifikace modelu

Verifikaci modelu můžeme rozdělit na (Marček, a další, 2008):

- a) Verifikaci statistickou
 - V této etapě se testuje koeficient determinace, významnost parametrů a koeficient autokorelace. Spadají zde také testy splnění předpokladů, potřebných k aplikaci konkrétních ekonometrických metod.
- b) Verifikaci ekonomickou
 - Zde se posuzuje ekonometrický model, v závislosti na informacích a závěrech z prvotní ekonomické teorie, aplikované na dané podmínky zkoumaného systému.

2.1.6.4 Aplikace modelu

Ekonometrický model je možno využít třemi způsoby:

- a) Využití modelu pro analýzu minulého vývoje. Můžeme tedy modelem vytvářet simulaci, u které ověřujeme například přesnost, nebo adekvátnost vývoje ekonomických veličin.
- b) Využití modelu při prognózách. Prognostická aplikace pak umožňuje s velkou pravděpodobností určovat vývoj ukazatelů a předpovídat důsledky změn různých opatření ekonomické politiky.

- c) Využití modelu pro výběr variant ekonomických nástrojů řízení. Při této aplikaci modelu se hledají takové hodnoty vysvětlujících proměnných, které vedou k určitým hodnotám vysvětlovaných proměnných.

Ekonometrické modelování je také možno kombinovat s jinými modelovacími technikami. A to hlavně optimalizačními a aktuálně také s metodami a technologiemi, jež spadají do oblasti umělé inteligence (Marček, a další, 2008).

2.1.7 Box-Jenkinsova analýza dat, ARIMA procesy

ARIMA tedy Auto Regressive Integrated Moving Average modely se zakládají na předpokladu, že lze na hodnoty pozorování časové řady pohlížet jako na realizaci náhodných procesů. Jednotlivé proměnné a jejich závislosti, pomocí kterých jsou hodnoty časové řady vytvořeny, se nehledají mimo časovou řadu. Nicméně se metodami statistické analýzy vyhledávají takové časové posuny a závislosti v jednotlivých hodnotách časové řady, které adekvátním způsobem odpovídají pozorování časové řady. Jde tedy o vyhledání vnitřní struktury a závislostí, které generují hodnoty časové řady s přijatelnou přesností. (Marček, a další, 2008) (Cipra, 2008)

Proces generování těchto dat je tvořen skupinou lineárních modelů, které se nazývají ARIMA. Podle (Marček, a další, 2008, s. 25) je v ekonomické praxi mnohdy jedinou možností jak vyhledat závislosti ve vývoji časové řady. Tato skutečnost je dána tím, že v praxi je mnohdy velmi obtížné získat relevantní data, která ovlivňují daný ekonomický proces, nebo není k dispozici vhodná ekonomická teorie či hypotéza, pomocí které bychom tyto data určili.

Výhody B-J metodologie (Arlt, a další, 2002):

- dobrá flexibilita a rychlá adaptace na změnu charakteru modelovaného procesu,
- v mnoha případech dává nejlepší výsledky, vzhledem k MRSE.

Nevýhody B-J metodologie:

- pro její zavedení je potřebná dostatečně dlouhá časová řada,
- obtížná možnost interpretace výsledného modelu. Velmi těžko se přesvědčuje oponenta modelu, že lze modelovat pouze pomocí zpožděné časové řady a náhodných šoků.

Autoregresní procesy AR(p)

ARMA procesy jsou procesy s nekonečným počtem AR a MA členů. V reálných situacích však počítáme se skupinou členů AR a MA v jejich konečných variantách. Místo počtu AR a MA členů využívají stupně modelu. Pro AR procesy je to stupeň „p“ a pro MA proces stupeň „q“. Podle (Marček, a další, s. 31) se to zdůvodňuje pravidlem parsimony, podle kterého postačuje navrhovat modely se počtem stupňů p a q, které s dostatečnou přesností reprezentují reálný modelovací proces s relativně málo členy (Marček, a další, 2008).

Nejjednodušším modelem ARIMA procesu je autoregresní model stupně p. V tomto modelu je pozorovaná hodnota y_t lineárně závislá pouze od předcházejícího p pozorování a náhodného členu ε_t .

Autoregresní model prvního stupně tedy můžeme zapsat následující rovnicí:

$$y_t = \xi + \phi_1 y_{t-1} + \varepsilon_t \quad (2.5)$$

V tomto zápise pak:

y_t - představuje závislou proměnnou

y_{t-1} - představuje o jedno období posunutou proměnnou y_t

ϕ_1 - představuje autoregresní koeficient

ε_t - představuje náhodnou složku

ξ - představuje konstantu modelu

Pomocí zpětnovazebného operátoru B můžeme tento model zapsat následujícím způsobem:

$$\phi_1(B)y_t = \xi + \varepsilon_t \quad (2.6)$$

Autoregresní model libovolného stupně p by tedy měl následující zápis:

$$y_t = \xi + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t \quad (2.7)$$

AR proces p stupně s sebou nese následující vlastnosti:

- a) AR proces je vždy invertovatelný
- b) Střední hodnota procesu je nulová

Procesy klouzavých průměrů MA(q)

Autoregresní model vytváří předpověď na základě konečného počtu předcházejících hodnot naměřené proměnné. Proces klouzavých průměrů MA sestavuje předpověď na základě konečného počtu minulých chyb v odhadu. (Marček, a další, 2008)

Zápis ARMA modelu pouze pomocí MA částí procesu, čili jen pomocí q členů, můžeme tento model zapsat následujícím způsobem pro MA(1):

$$y_t = \mu + \varepsilon_t - \theta_1 \varepsilon_{t-1} \quad (2.8)$$

Kde:

ε_t – představuje v čase nekorelované náhodné hodnoty

ε_{t-1} – představuje o období posunutou náhodnou složku

θ_1 – představuje koeficient modelu

μ – představuje průměrnou hodnotu proměnné y_t

Pro model MA(q) je tedy možno vytvořit následující zápis:

$$y_t = \mu + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_q \varepsilon_{t-q} \quad (2.9)$$

Vlastnosti MA(q) procesu:

- a) Má nulovou střední hodnotu
- b) Proces je stacionární pro libovolnou volbu parametru

Smíšené procesy ARMA(p,q)

Analýzy reálných časových řad velmi často vede na procesy ve kterých se vyskytuje jak autoregresní část, tak část klouzavých průměrů. Vytváří se tedy smíšené ARMA(p, q) procesy s oběma stupni (p, q). (Marček, a další, 2008)

Všeobecný zápis smíšeného modelu ARMA(p,q) tedy zapisujeme následující rovnicí:

$$y_t = \xi + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_q \varepsilon_{t-q} + \varepsilon_t \quad (2.10)$$

Kde:

$y_{t-1}, y_{t-2}, y_{t-p}$ – představují časově posunuté proměnné

ϕ_1, ϕ_2, ϕ_p – představují autoregresní koeficienty

$\theta_1, \theta_2, \theta_q$ – představují koeficienty modelu klouzavých průměrů

$\varepsilon_{t-1}, \varepsilon_{t-2}, \varepsilon_{t-q}$ – představují časově posunuté náhodné složky

Základní vlastnosti ARMA(p,q):

- a) Má nulovou střední hodnotu
- b) Autokorelační funkce je ve tvaru lineární kombinace klesajících geometrických posloupností s geometricky klesající amplitudou po prvních (p,q) hodnotách
- c) Parciální korelační funkce je ve tvaru lineární kombinace klesajících geometrických posloupností s geometricky klesající amplitudou po prvních (p,q) hodnotách.

Pro ARMA(p,q) modely pak platí aplikovatelnost stacionárních časových řad. Pokud je tedy vstupní časová řada analyzovaná jako řada nestacionární, uplatňujeme na ně metody transformace časové řady do její stacionární podoby. (Marček, a další, 2008)

Těmito metodami mohou být:

- a) Metody odstranění trendové složky
- b) Transformace časové řady pomocí jejího zlogaritmování
- c) Transformace pomocí diferenciací časové řady

V momentě nutnosti upotřebení metody transformace časové řady pomocí diferenciací, je možné tuto metodu uplatnit přímo na ARMA(p,q) model. Pro tento účel je definován operátor diferencí:

$$\Delta y_t = y_t - y_{t-1} \quad (2.11)$$

Výraz Δy_t tedy vyjadřuje novou hodnotu časové řady. Jak lze z dané rovnice 2.11 vidět, každý stupeň difference nám o daný stupeň snižuje délku časové řady. Diferencovaný ARMA(p,q) proces je pak transformován na proces ARIMA(p,d,q), kde stupeň „d“ označuje stupeň difference časové řady. (Marček, a další, 2008) (Hančlová a Tvrđý, 2003)

Modelování časových řad pomocí ARMA procesů

Modelování ARMA procesů je možné stanovit do čtyř po sobě jdoucích kroků:

- a) Identifikace modelu
- b) Odhad parametrů
- c) Diagnostická kontrola
- d) Prognózování

Identifikace modelu

V etapě identifikace se zpravidla odstraňuje trendová složka modelu a model se předběžně identifikuje. Tento model může být pouze autoregresivní - AR(p) nebo pouze model klouzavých průměrů MA(q). Také může dojít k situaci, kdy bude model smíšený. ARMA(p,q).

Tyto modely mohou nabývat různých hodnot svých stupňů (p,q). Proces identifikace se pak nejčastěji vykonává pomocí zkoumání výběrové autokorelační funkce a výběrové parciální autokorelační funkce analyzované časové řady. (Marček, a další, 2008)

Rovnice pro výběrovou autokorelační funkci – ACF, má následující tvar:

$$r_k = \frac{\sum_{t=1}^{N-k} (y_t - \bar{y})(y_{t+k} - \bar{y})}{\sum_{t=1}^N (y_t - \bar{y})^2}, \quad k = 0, 1, \dots, K. \quad (2.12)$$

Pro výběrovou parciální autokorelační funkci – PACF transformujeme rovnici pomocí Yule-Walkerovými rovnicemi na tvar rovnice:

$$r_j = \hat{\phi}_{k1} r_{j-1} + \hat{\phi}_{k2} r_{j-2} + \dots + \hat{\phi}_{kk} r_{j-k}, \quad \text{pro } j = 1, 2, \dots, k. \quad (2.13)$$

Po výpočtu hodnot ACF a PACF z výběrového souboru dat můžeme tyto hodnoty zobrazit graficky. Pomocí grafického vývoje těchto hodnot jsme schopni určit předběžný typ ARMA(p,q) procesu. V tabulce 2.1 jsou určené charakteristiky teoretického vývoje hodnot ACF a PACF a jejich vlivu na typ ARMA modelu a jeho stupně (p,q). (Marček, a další, 2008)

Tabulka 2.1 Charakteristiky vývoje hodnot ACF a PACF a jejich vliv na model

ACF	PACF	Model
Pozvolně utlumovaná	Náhle utlumená po posunu p	AR(p)
Náhle utlumená po posunu q	Pozvolně utlumovaná	MA(q)
Pozvolně utlumovaná	Pozvolně utlumovaná	ARMA(p,q)

Ve skutečnosti se však mohou vyskytovat další kombinace těchto průběhů. Řešení toho jaký model zvolit a s jakými stupni pak poskytují některé numerické algoritmy či rozhodovací kritéria. Patří mezi ně především AIC kritérium – tedy Akaike Information Criterion. V případě stanovování jednotlivých modelů pak volíme takový model, jehož AIC je nejnižší. (Marček, a další, 2008)

Odhad parametrů modelu

Po identifikaci modelu ARMA se tedy určí jeho typ a stupeň. Z identifikace tedy může plynout použití AR, MA nebo ARMA (v případě potřeby difference dat ARIMA) model. Určeným typem modelu je určený také charakter modelu z hlediska jeho parametrů. AR model je lineární. Model MA a ARMA je model nelineární. Charakter modelu určuje odhadové metody pro odhad parametrů modelu. Nejvíce používanou metodou na odhad parametrů

nelineárních modelů je metoda maximální věrohodnosti. Tato metoda je iterativní a vyžaduje tedy počáteční odhad parametrů. Tento odhad se vykonává ve dvou etapách. V prvním kroku se odhadnou počáteční hodnoty parametrů. V druhém kroku se tyto hodnoty z prvního kroku použijí na iterativní vyčíslení konečných hodnot parametrů. (Marček, a další, 2008)

Diagnostická kontrola

Diagnostická kontrola modelu zahrnuje dva přístupy. Tím prvním je kontrola koeficientu determinace. Druhý krok zahrnuje kontrolu reziduí.

Koeficientem determinace je vyjádřena míra závislosti endogenní proměnné na exogenních proměnných. Je to tedy míra, který vyjadřuje stupeň vysvětlení proměnlivosti endogenní proměnné daným regresním modelem. Tento koeficient by měl být co nejvyšší. Jeho maximální hodnota je $=1$ (tedy 100 %).

Testování reziduí vychází z předpokladu, že rezidua by měla být časově stabilní, náhodné proměnné s přibližně normálním rozdělením s nulovou střední hodnotou a konstantním rozptylem. Tyto vlastnosti reziduí pak znamenají, že model použitý pro konstrukci prognóz musí být na tyto vlastnosti reziduí testovaný. Adekvátnost modelu typu ARMA pak vede k jeho schopnosti generovat data blízké datům pozorovaným a rozdíly (tedy chyba), mezi těmito daty musí být bílým šumem. (Marček, a další, 2008)

Hodnocení prognóz modelu

Celková časová řada je na začátku rozdělena do trénovací (kvantifikační) a validační období. Validační období pak poskytuje možnost porovnat výsledky prognóz modelu s již naměřenými daty. Trénovací část časové řady je delší než část validační a odhad parametrů modelu je založen pouze na datech z předložené trénovací části.

Pro hodnocení modelu z hlediska prognostiky se používá vícero charakteristik. Tyto hodnoty se pak dají použít pro porovnávání predikční přesnosti mezi několika druhy přístupů k prognózování. (Marček, a další, 2008)

Nejčastější charakteristiky pro hodnocení prognóz jsou:

- a) Mean Error – tedy průměrná chyba prognózy
- b) Mean Square Error – průměrný čtverec chyby
- c) Root Mean Square Error – odmocněný průměrný čtverec chyby
- d) Mean Absolute Error – průměrná chyba absolutních hodnot

Pro porovnávání prognostických přesností mezi prognózami statistických modelů ARIMA a prognózami neuronových sítí bude použita charakteristika RMSE (Root Mean Square Error).

Vzorec pro výpočet této charakteristiky je následující:

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (\bar{y}_t - y_t)^2}{n}} \quad (2.14)$$

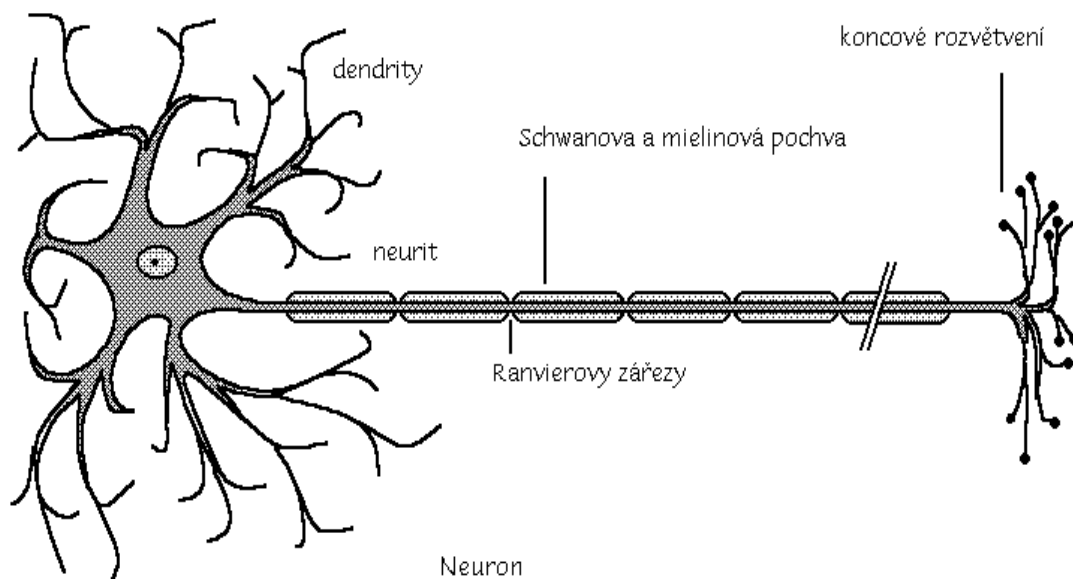
2.2 Neuronové sítě

Umělé neuronové sítě se v současné době používají na řešení mnoha problémů, jež jsou velmi vzdálené problémům, pro které byly tyto sítě původně vyvinuté. Umělé neuronové sítě vznikly po vzoru nervového systému. Jsou tedy původně vyvinuté podle mozkových neuronů a kopírují jejich biologické funkce. Od svého vzniku prošly umělé neuronové sítě teoretickým i aplikačním vývojem. V devadesátých letech byly umělé neuronové sítě úspěšně nasazeny například v oblasti managementu, kde úspěšně predikovaly krach banky. Dále byly nasazeny v oblasti telekomunikací a v oblasti modelování časových řad, jako alternativa statistických metod. Umělé neuronové sítě mohou být aplikovány jako univerzální aproximátor. Mohou tedy aproximovat libovolnou spojitou funkci, za předpokladu kvalitního vzoru učení. (Marček, 2013) (Kačenka, 1998)

2.2.1 Zpracování informací neuronovou buňkou

Elementárním stavebním prvkem nervového systému je neuronová buňka. Neuronová buňka zpracovává různé vstupní informace a dále komunikuje s různými částmi těla.

Lidský mozek obsahuje přibližně 10^{11} neuronových buněk v různých typech. Nákres jednoduché mozkové neuronové buňky je znázorněn na obrázku 2.1. (Marček, 2013)



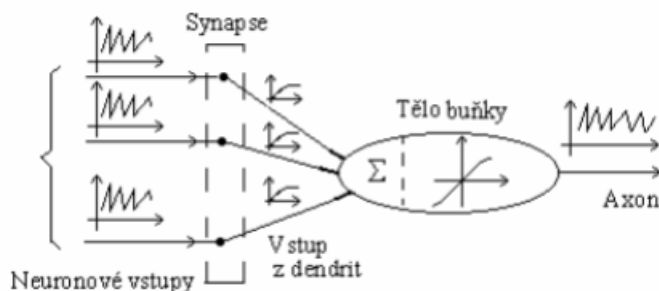
Obrázek 2.1 Biologický neuron (zdroj: gyza.cz)

Spojení několika neuronových buněk se pak nazývá neuronová síť. Neuronové tkanivo je tvořeno z dendrit, které jsou spojeny s tělem buňky. Jedno dlouhé tkanivo se pak nazývá axón. Pomocí axónu se pak neuronové buňky větví a vzájemně propojují. Na konci těchto tkání se nacházejí synaptické spojení, kde dojde s informačním spojením s jinou buňkou. Pro zpracování a uchování jednotlivých informací jsou důležité následující části (Marček, 2013):

- a) Synaptické spojení – ty jsou chápány jako místa příjmu informací a místo pro uchování dat z jiných neuronů
- b) Tělo buňky (perikarion, sóma) – zde dochází ke zpracování a přijímání informace ze synaptických spojení
- c) Axón – který přenáší zpracované informace pro další neurony

Přenos jednotlivých signálů v rámci neuronových buněk je složitý chemický proces. Neuronové buňky jsou obklopené chemickými ionty alkalických kovů a dalších látek. Jsou mezi nimi zastoupeny zejména ionty vápníku, draslíku a sodíku. Tyto látky jsou schopny vytvářet aktivní neuronové odezvy, které se nazývají akční potenciály, či nervové impulzy. Vznik a přenos akčního potenciálu způsobuje záměna iontů draslíku a vápníku, která je vyvolána změnou propustnosti buňkové membrány. (Marček, 2013)

Celkový funkční proces cesty signálu v rámci biologických neuronových sítí se dá shrnout následujícím způsobem. Pokud jsou nervové impulzy ve formě akčních potenciálů dopraveny pomocí synaptických spojení, jsou propuštěny do dendrit. Toto má za následek vznik elektrické reakce (vzruchu). Tato elektrická reakce může být excitační nebo inhibiční. Tedy zesilující nebo tlumící, v závislosti na jednotlivých membránách dendritů. Velikost signálu v dendritu je proporcionální průměrné frekvenci, s kterou impulzy dosahují synaptické spojení. Excitace nebo inhibice potenciálu signálu má pak za účinek dosažení určité úrovně potenciálu. Pokud tento potenciál přesáhne určitou úroveň, neuronová buňka se vybudí (aktivuje). Pokud je neuronová buňka aktivována, pak vypudí signál do axónu a skrze axón pak putuje tento signál do jiných neuronů. Toto funkční spojení je pak možné schématicky znázornit pomocí modelu z obrázku 2.2. V návaznosti na tento model vznikl McCulloch-Pits umělý neuron. (Marček a Marček, 2006)



Obrázek 2.2 Funkční model zpracování signálů v biologickém neuronu (zdroj:(Marček, a další, 2006, s. 13)

Z hlediska zpracování signálů v umělém neuronu pak jde o jeho transformaci. Transformace signálu není samoúčelná, ale je závislá na určitém cíli. Jde tedy o vyhledání vstupně-výstupní funkce s parametry, které povedou k požadovaným vlastnostem na výstupu. Například z hlediska predikční činnosti neuronu je účelem vyhledat funkci s parametry vedoucí k minimalizování chyby v dané predikci. Samotný proces vyhledávání vstupně-výstupní funkce se pak nazývá identifikace procesu. (Marček a Marček, 2006)

2.2.2 Vícevrstvé neuronové sítě

Nejrozšířenější architektura vedoucí k propojení jednotlivých neuronů se nazývá vícevrstvá (nebo také vrstvená) neuronová síť. Tato architektura je pak tvořena nejméně třemi vrstvami neuronů, přičemž každá vrstva nese svá specifika. Každý typ neuronové sítě, si pak ještě tuto architekturu upravuje podle svých specifik. (Marček, 2013) (Šíma a Neruda, 1996)

Vrstvy, ze kterých je vícevrstvá neuronová síť složena je možné rozdělit do tří skupin.

- 1) Vstupní neuronová vrstva – tato vrstva poskytuje síti neurony, které slouží pro přijímání vstupních dat do sítě. Z této vrstvy jsou pak data šířeny dále. Počet neuronů v této vrstvě je teoreticky od jednoho po nekonečný počet.
- 2) Skrytá neuronová vrstva – tato vrstva se teoreticky může ve vícevrstvé architektuře objevit v nekonečném počtu. Skrytá vrstva slouží jako hlavní výpočetní složka sítě. Počet neuronů v této vrstvě může teoreticky nabývat od jednoho po nekonečný počet neuronů. Nicméně pro výsledky sítě v predikčních systémech neplatí, že čím vyšší počet neuronů této vrstvy, tím přesnější výsledky. Určení toho, kolik se má v dané architektuře objevit neuronů střední vrstvy je jedním z největších problémů neuronových sítí. V predikčních systémech se používá architektura s jednou skrytou vrstvou.
- 3) Výstupní neuronová vrstva – vrstva sloužící pro sumarizaci výsledku neuronové sítě. Teoretický počet neuronů v této vrstvě je opět od jedné k nekonečnému počtu neuronů, nicméně v predikčních systémech se používá jeden výstupní neuron.

Co se týče celkového chodu sítě, můžeme tento proces rozdělit na dvě hlavní složky. Konkrétně dopředný a zpětný chod. Oba tyto chody jsou opět závislé na konkrétním typu sítě. Tedy síť typu perceptron bude mít specifický a odlišný typ obou chodů oproti síti typu RBF. Nicméně lze dopředný i zpětný chod vícevrstvé architektury shrnout v následujících bodech.

- 1) Synaptické váhy sítě jsou nastaveny na náhodnou úroveň
- 2) Skrze vstupní neurony je do sítě přivedena datová složka
- 3) Tyto data jsou pomocí vazeb přivedeny neuronům střední vrstvy a jejich hodnota je upravena složkou synaptických vah. Tato složka je pak přivedena jako vstup neuronům střední vrstvy.
- 4) Neurony střední vrstvy provedou sumarizaci dat a podle své aktivační funkce jsou excitovány na úroveň danou touto funkcí. Tento postup následně provádí všechny skryté vrstvy
- 5) Poslední skrytá vrstva dodá data výstupní vrstvě. Tato vrstva pak provede sumarizaci a na výstupu poskytuje výsledek.

Vzhledem k tomu, že přímý výpočet hodnoty synaptických vah je v praxi nevhodný, poskytuje síti možnosti změny těchto vah zpětný chod. Zpětný chod jinými slovy poskytuje metody učení sítě. Nejužívanější metodou je back-propagation algoritmus, který ve zpětném chodu šíří zpětně chybu. K celkovému chodu back-propagation algoritmu je potřebná pouze tato chyba a příslušná vstupní datová položka. Změna vah v tomto algoritmu není okamžitá, ale provádí se postupně. (Marček a Marček, 2006)

2.2.3 Neuronová síť Perceptron

Perceptron je speciálně navržený formální neuron. Perceptron představuje všeobecný výpočtový prvek neuronových sítí, ve kterých je vnitřní potenciál neuronu počítaný jako vážený součin vstupních dat a příslušných vah (Marček, 2013) (Kecman, 2001).

Matematickým zápisem potom vnitřní potenciál perceptronu můžeme vyjádřit takto:

$$\xi = \sum_{i=0}^n x_i w_i \quad (2.15)$$

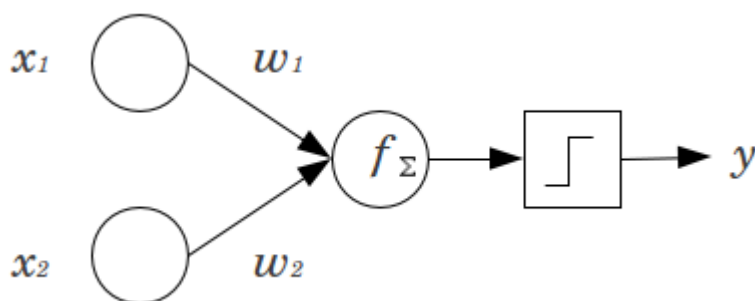
Kde:

ξ – představuje sumu násobků dvojic vstupních dat a příslušných vah

x_i – je množina vstupních dat

w_i – je množina jednotlivých vah

Graficky je možno perceptron znázornit následovně pomocí obrázku 2.3.



Obrázek 2.3 Perceptron (zdroj: glowingpython.blogspot.cz)

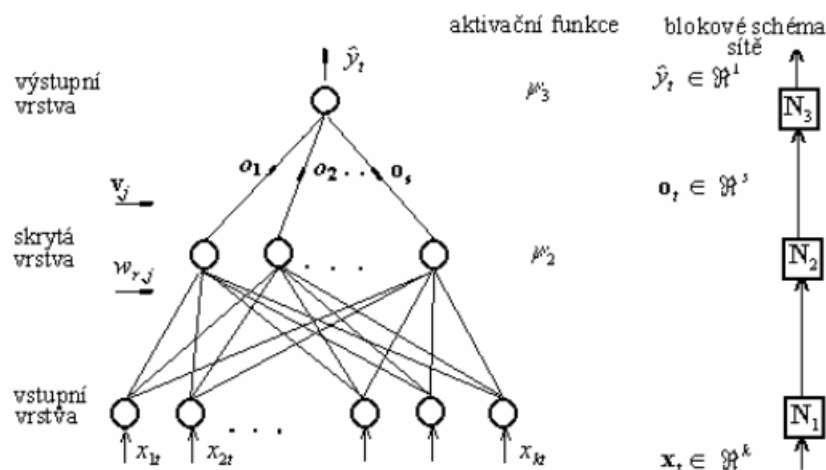
Perceptron je tedy neuron, který na vstupu přijímá vstupní data, které mohou nabývat jak reálných tak binárních hodnot a tyto vstupní data následně váží pomocí vah, které nabývají hodnot reálných čísel. Výstup perceptronu je následně možno zaspát následujícím způsobem (Marček a Marček, 2006):

$$y = f\left(\sum_{j=1}^{n+1} w_j x_j\right) \quad (2.16)$$

Funkce f je aktivační funkce neuronu. Tato funkce se následně liší v závislosti na použití dané neuronové sítě typu perceptron. V rámci použití perceptronové neuronové sítě pro predikci dat se využívá v neuronech skryté vrstvy Sigmoidní aktivační funkce.

Z hlediska složení neuronové sítě typu perceptron se používá vícevrstvá neuronová síť s neurony střední vrstvy typu perceptron. Z hlediska vícevrstvé sítě typu perceptron je zejména důležité zmínit to, jakým způsobem se tato síť učí, tedy jakým způsobem mění své váhy tak, aby dosahovala co nejvyšší prognostické přesnosti. (Marček a Marček, 2006)

Pro potřeby popisu back-propagation algoritmu zobrazuje obrázek 2.4 architekturu perceptronové sítě s jednou skrytou vrstvou.



Obrázek 2.4 Architektura UNS s jednou skrytou vrstvou (zdroj: (Marček a Marček, 2006, s. 19))

Učení probíhá pomocí back-propagation algoritmu, který má pro vícevrstvou perceptronovou síť s jednou skrytou vrstvou následující postup. (Marček a Marček, 2006)

- 1) Po ukončení dopředného chodu je vypočítána chyba na výstupu podle vzoru

$$e = y - \hat{y} \quad (2.17)$$

- 2) Synaptické váhy neuronové sítě se adaptují tak, aby tuto chybu redukovaly. Problém nastává při rozdělení chyby mezi spodní a horní váhy neuronové sítě. Tento problém se řeší sestavením chybového termu. Chybový term horních vah v_j se počítá následovně

$$\Delta = e\psi_3(u), \text{ kde } u = \sum_{j=1}^s v_j o_j \quad (2.18)$$

Symbol s značí počet neuronů střední vrstvy. Změna vah pro aktualizaci jednotlivých synaptických spojení je dána gradientem chybové funkce výstupního neuronu, který je určen součinem parametru rychlosti učení, vstupu daného dopředného chodu a chybového termu, tedy $\eta o_j \Delta$

- 3) Pravidlo pro adaptaci vah v_j mezi neurony střední vrstvy a výstupním neuronem je:

$$v_j \leftarrow v_j + \eta o_j \Delta \quad j = 1, 2, \dots, s. \quad (2.19)$$

- 4) Pro adaptaci synaptických vah $w_{r,j}$, které se nacházejí mezi vstupní vrstvou a vrstvou skrytou, definujeme chybový term pro j -tý neuron skryté vrstvy podobně jako u neuronu výstupní vrstvy. Vychází ze z předpokladu, že každý neuron střední vrstvy nese určitou vinu na výstupu z výstupní vrstvy. Tato chyba se tedy proporcionálně váhami v_j přenáší na j -té neurony ve skryté vrstvě, tedy:

$$\Delta_j = \psi_2(\sum_{r=1}^k (w_{r,j} x_r)) v_j \Delta, \quad j = 1, 2, \dots, s. \quad (2.20)$$

Kde symbol k určuje počet neuronů vstupní vrstvy.

- 5) Pravidlo pro adaptaci samotných synaptických vah $w_{r,j}$ mezi vstupními neurony a neurony skryté vrstvy je:

$$w_{r,j} \leftarrow w_{r,j} + \eta x_r \Delta_j, \quad j = 1, 2, \dots, s, \quad r = 1, 2, \dots, k. \quad (2.21)$$

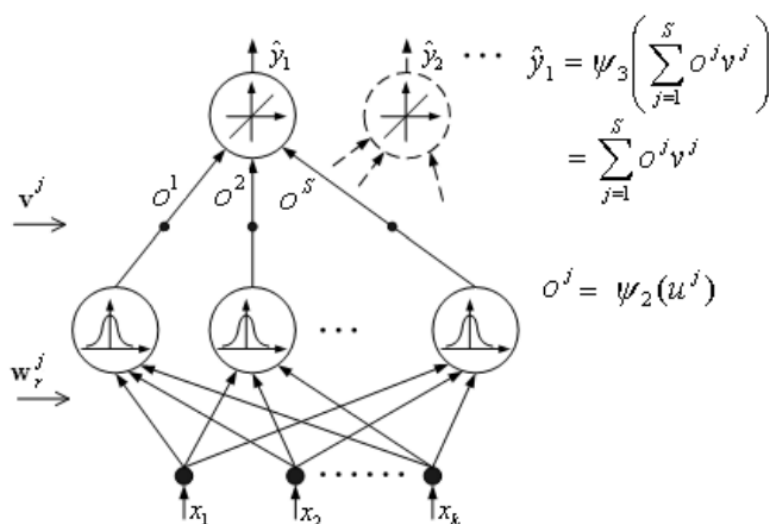
Tyto kroky se následně opakují po každém kroku dopředného chodu, čili po každém vpuštěném datovém řádku trénovací množiny.

Tento algoritmus může v praxi přinášet jedno úskalí. Tím je, že se v rámci trénování sítě nepodaří dosáhnout dostatečně malé chyby na výstupu. Důvodem je uvíznutí sítě v lokálním minimu. Tedy v bodě, kdy energetická hodnota funkce dál neklesá. Nejvhodnějším řešením tohoto problému je vhodná volba parametru rychlosti učení η . Rychlost učení nabývá hodnot z intervalu 0,1. Hodnota rychlosti učení s vyšší hodnotou v rámci intervalu může způsobovat oscilaci, naopak velmi nízká hodnota z intervalu bude způsobovat oscilaci. V praxi je tedy nutno vyzkoušet nastavení sítě s různými hodnotami tohoto parametru. (Marček a Marček, 2006)

2.2.4 RBF sítě

RBF sítě mají význačné postavení mezi sítěmi s dopředným šířením signálu, vzhledem ke svým schopnostem aproximovat vstupně-výstupní funkce a schopnostem klasifikačním. Neurony skryté vrstvy RBF sítě jsou aktivované pomocí funkcí Gaussovského typu, ty se taky označují jako Radial Basic Function. Odtud tedy název sítě RBF. Největšího rozmachu zaznamenaly RBF sítě v 80 a 90 letech minulého století. (Marček, 2013) (Vondrák, 2002)

Velkou předností RBF sítě je rychlá schopnost učení sítě a její efektivita. Aproximace vstupně-výstupní funkce má u RBF sítě lokální charakter, data se modelují jen v nejbližším okolí jejich středů. Architektura RBF sítě je patrná na obrázku 2.5.



Obrázek 2.5 Topologie RBF sítě (zdroj: (Marček a Marček, 2006, s. 170))

Co se architektury sítě týče, RBF sítě se podobají sítím typu perceptron s jednou skrytou vrstvou. Jak síť typu RBF tak síť typu perceptron mají jednu skrytou vrstvu, v jejichž neuronech je nelineární aktivační funkce. Je zde však několik rozdílů:

- Aktivační funkce ve výstupním neuronu sítě RBF je vždy lineární (síť typu perceptron nevyžaduje lineární funkci)
- Aktivační funkce v neuronu střední vrstvy je funkce Gaussovského typu

Nejpodstatnějším rozdílem sítě RBF oproti vícevrstvé síti typu perceptron je způsob výpočtu potenciálu neuronů skryté vrstvy. Zatímco v sítích perceptron je potenciál počítán jako skalární součin vstupního vektoru a vektoru vah, je potenciál v sítích typu RBF počítán jako čtverec rozdílu vzdálenosti mezi koncovými body vstupního vektoru a vektoru vah. (Marček a Marček, 2006)

V matematickém zápisu tedy:

$$u^j = ||x - w^j||^2, \quad j = 1, 2, \dots, s. \quad (2.22)$$

Vektor vah w^j neuronů skryté vrstvy je chápán jako střed specifické aktivační funkce, většinou tedy Gaussovské funkce. Výpočet výstupu z neuronů skryté vrstvy je počítán následujícím způsobem:

$$\psi_1(u^j) = \exp\left(-\frac{u^j}{2\sigma_j^2}\right) = \exp\left(-\frac{||x - w^j||^2}{2\sigma_j^2}\right) \quad (2.23)$$

Kde:

σ_j^2 - znamenají rozptyly jednotlivých prvků vstupního vektoru x .

Adaptace parametrů RBF sítě

Dopředné šíření signálu v rámci RBF sítě se provádí následujícím způsobem:

- 1) Na vstup neuronové sítě se předloží vstupní vektor trénovací množiny, tedy jeho prvky $x_t = (x_{1t}, x_{2t}, \dots, x_{kt})$, kde index t znázorňuje t -tý prvek trénovací množiny.

- 2) Vypočítá se potenciál neuronu podle vzorce 2.22 a výstup z neuronu střední vrstvy podle vzorce 2.23.
- 3) Výstup z neuronu střední vrstvy je přiveden na vstup do neuronu výstupní vrstvy. Zde se výstupní hodnota spočítá pomocí lineární aktivační funkce, tedy skalárním součinem vstupu a příslušných vah
- 4) Nakonec je spočítána chybová hodnota e_t jako $e_t = y_t - \hat{y}_t$

Zpětné šíření chybové hodnoty, pro změnu vah výstupního neuronu se provádí pomocí následující formule:

$$v_j^t \leftarrow v_j^t + \eta e_t o_t^j \quad (2.24)$$

Pomocí back-propagation algoritmu lze měnit také hodnoty vah do neuron skryté vrstvy, nicméně v rámci aproximace vstupně-výstupních funkcí lze použít pro hodnoty vah pro neurony skryté vrstvy shlukový algoritmus.

Jakákoli množina bodů může být reprezentována shluky dat, které jsou soustředěné okolo středu shluku. Tyto středy shluků, konkrétně jejich souřadnice jsou pak použity jako hodnoty vah pro skrytou vrstvu RBF neuronu. Nejpoužívanější metodou pro vyhledávání těchto shluků je metoda K-means clustering algorithm. (Marček a Marček, 2006) (Marček, 2013)

Zadaný počet shluků, pak reprezentuje počet neuronů stření vrstvy RBF sítě. K-means clustering algorithm se dá zapsat následujícími kroky:

- 1) Zvolí se počet potřebných shluků, čili výsledný počet neuronů střední vrstvy a hodnota rychlosti učení. Ta je běžně nastavována od $<0;1>$.
- 2) Těmto centrům se počátečně nastaví náhodná hodnota, v praxi se používá prvotní umístění shluků do již platného souřadnicového systému, jež vytváří dané vstupní data (minimalizuje se tak čas potřebný k chodu algoritmu)
- 3) Na vstup se předloží t-vstup ze vstupního souboru s označením x_t
- 4) Vyhledá se vítězný shluk, tedy ten, jehož poloha je k danému bodu nejbližší ze všech daných shluků

- 5) Vypočítá se nová hodnota polohy shluku v souřadnicovém systému pomocí:

$$c_j^{new} = c_j^{old} + \eta(x_t - c_j^{old}) \quad (2.25)$$

Kde:

η - je parametr rychlosti učení

- 6) Kroky 3 až 5 se opakují pro zadaný počet epoch, přičemž jedna epocha reprezentuje vstup celého vstupního souboru do algoritmu. Po čas běhu algoritmu se po každé epoše upraví rychlost vah tak, aby se postupně snižovala až na hodnotu nulovou.

To má za následek, že v prvotních epochách jsou shluky rozmisťovány velkou rychlostí a jejich poloha je poté jemně laděna na konečnou hodnotu

- 7) Pro každou hodnotu shluku se na konci algoritmu uloží, kolik bodů ze vstupního souboru tento shluk reprezentuje a vypočítá se jeho rozptyl.

Na konci algoritmu je tedy k dispozici řada shluků, které s sebou nesou hodnoty svých souřadnic. Tyto souřadnice jsou poté načteny jako hodnoty vah pro neurony střední vrstvy RBF sítě. K dispozici je také hodnota rozptylu daného shluku, která je potřebná jako vstupní proměnná do aktivační funkce neuronu střední vrstvy. V praxi se pak používá průměrný rozptyl, vzniklý z množiny rozptylů jednotlivých shluků, či se nastavuje zvětšení těchto hodnot tak, aby se vzájemně překrývaly.

Učení RBF sítě je tak rychlejší oproti učení perceptronu. Není zde třeba již měnit váhy neuronů střední vrstvy. Mění se pouze váhy neuronů výstupní vrstvy. (Marček a Marček, 2006)

Co se týče hodnocení predikční přednosti, je opět využívána minimalizace průměrné chyby odhadu MSE, popřípadě její varianty, tak jako u sítě perceptron. Učení sítě RBF se tak stejně jako u perceptronu snaží tuto hybu minimalizovat.

2.2.5 Rozdělení vstupního souboru

Pro neuronové sítě využívající učení s učitelem je nutnost využití vstupního souboru. Tento vstupní soubor se pak pro potřeby sítě rozděluje na dvě množiny. Množinu trénovací a množinu validační.

V rámci trénovací množiny má neuronová síť k dispozici vstupní data a k nim správný, tedy naměřený výstup. Důležité je to, že neuronová síť se v rámci této množiny může učit tak, aby minimalizovala chybu na výstupu.

Validační množina taktéž přináší vstupní data a k nim naměřený výstup, nicméně neuronová síť se již na této množině neučí. Validační data jsou pak využita pouze pro měření chyby prognózy sítě.

Rozdělení vstupního souboru na tyto dvě množiny je také důležité z důvodu možnosti přeučení sítě. Tento jev nastává, pokud je síť přetrénovaná daty trénovací množiny a tyto data pak dokáže velmi přesně predikovat, nicméně síť později není schopna generalizace. Odstranění tohoto jevu docílíme právě rozdělením vstupního souboru, neboť učení sítě je ukončeno tam, kde je nejnižší chyba prognózy na validační množině. (Marček a Marček, 2006)

3 Návrh a vytvoření simulátoru pro simulování činností neuronových sítí

V této kapitole je nastíněna problematika návrhu a tvorby nástroje pro simulování chodu neuronových sítí typu RBF a neuronových sítí typu Perceptron

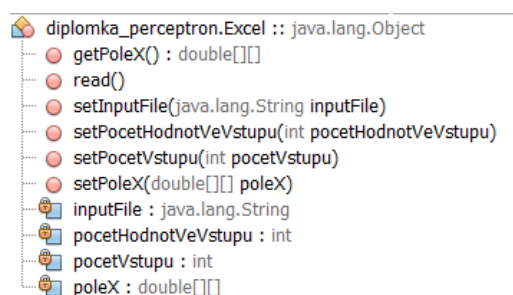
Simulátor je vytvořen v rámci jazyka JAVA, v programovacím prostředí NetBeans v jeho nejnovějším vydání 8.0.2. Vzhledem k tomu, že se architektury neuronových sítí RBF a Perceptron liší a přinášejí tedy i jiný pohled na algoritmizační řešení jejich výpočtů, byly od sebe tyto sítě programově odděleny. Každá architektura je tedy dostupná zvlášť v oddělených projektech.

3.1 Načítání dat

Sítě mají společné řešení načítání vstupních dat. Tento problém je vyřešen pomocí knihovny JXL. Tato knihovna přináší právě programové řešení načítání dat z datových listů programu MS Excel. Každý projekt jednotlivé neuronové sítě, tak má tuto knihovnu k dispozici, spolu s přednastavenou třídou Excel, jež programově načítání dat z excelovských listů zabezpečuje. Načítání dat však má určitá pravidla, a tak je nutno datový soubor se vstupními daty pro neuronovou síť upravit jednotným postupem.

- Datový soubor neobsahuje textové názvy sloupců, první řádek již obsahuje numerickou hodnotu vstupu
- Numerické hodnoty vstupů mají jak desetinný oddělovač tečku
- První sloupec datového souboru je vždy výstupní hodnotou datového souboru, připravený jako složka učitele
- Navazující sloupce jsou již připraveny jako vstupní hodnoty.

Kompletní pohled na metody, které obsahuje třída Excel, zobrazuje obrázek 3.1.



Obrázek 3.1 Metody třídy Excel

Datový soubor je pak možno umístit kdekoli v rámci pevného disku výpočetní jednotky, nad kterou je program spuštěn. Pokud je datový soubor umístěn přímo do složky programového projektu, je načten automaticky. Pokud je umístěn v jiné části, je nutno jeho cestu umístění zadat.

3.2 Nástroj simulace neuronové sítě typu perceptron

Simulace běhu neuronové sítě typu perceptron je realizovaná programovou třídou jménem Perceptron. Tato třída pak svými metodami algoritmicky řeší kompletní chod architektury sítě typu Perceptron, a to jak trénování sítě, tak její následnou predikci nad validační množinou.

Jako vstupní parametry pro síť perceptron slouží následující parametry:

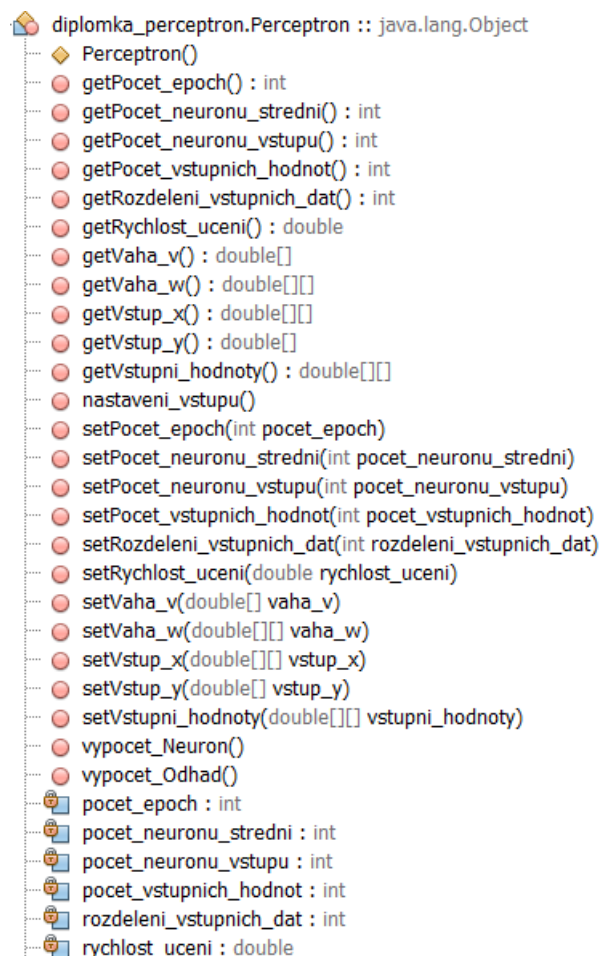
- a) Počet epoch – vyjadřuje počet epoch učení sítě
- b) Počet vstupních hodnot – vyjadřuje celkový počet vstupních hodnot, tedy počet řádků vstupů v datovém souboru
- c) Počet hodnot pro trénování – vyjadřuje kolik hodnot z celkového datového souboru, bude použito pro trénování sítě. Rozdílem hodnot počet vstupních hodnot a počet hodnot pro trénování vzniká počet datových záznamů pro validační množinu
- d) Počet neuronů střední vrstvy – určuje, kolik neuronů má být v architektuře použito v rámci střední (skryté) vrstvy sítě
- e) Rychlost učení – číslo přímo určující jak rychle se má daná síť učit

S těmito údaji si na vstupu nástroj vystačí a je schopen již sám bez dalších zásahů provést vytvoření sítě a její následný výpočet. Pokud by však vyvstala nutnost dalších vstupů do architektury sítě, je možno velmi jednoduše měnit taktéž další přenastavení architektury, jako například změnu typu aktivačních funkcí apod.

Kromě jiných podpůrných metod třídy Perceptron je její stěžejní částí metoda `vypocet_Neuron` a třída `vypocet_Odhad`. Tyto metody pak přímo řeší aplikaci neuronové sítě nad trénovací a validační množinou tak, jak je její architektura popsána v teoreticko-metodologické části práce. Celkově je programová struktura nástroje pro perceptronovou síť umístěna na zhruba tři sta osmdesáti řádcích kódu.

Kompletní pohled na jednotlivé metody a proměnné třídy Perceptron poskytuje obrázek

3.2.



Obrázek 3.2 Metody a proměnné třídy Perceptron

3.3 Nástroj simulace neuronové sítě typu RBF

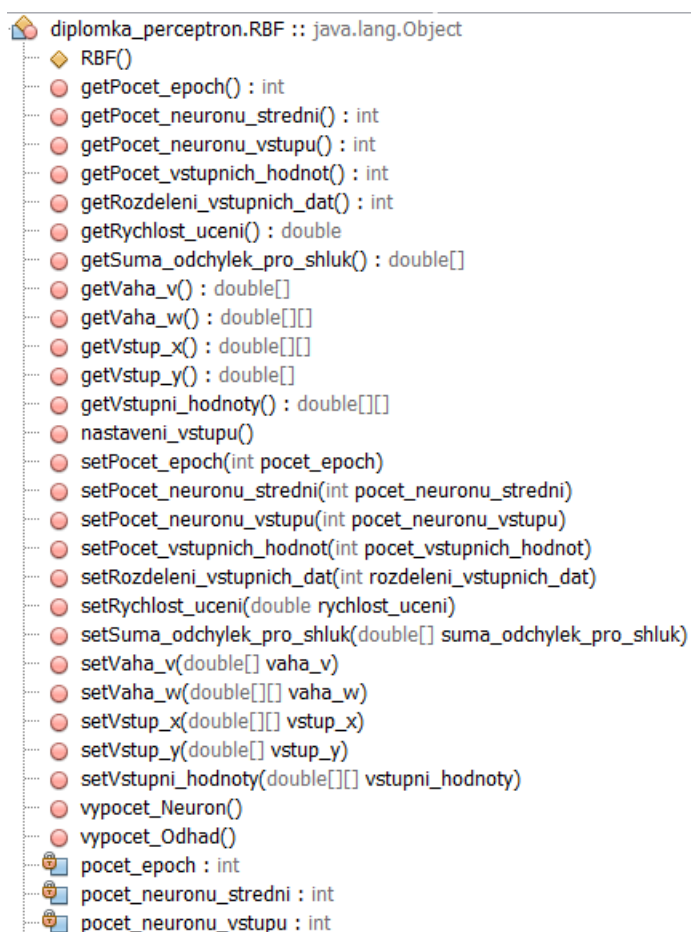
Simulace běhu neuronové sítě typu RBF je realizována programovou třídou jménem RBF. Tato třída kompletně řeší chod tohoto typu neuronové sítě. Vstupní parametry této třídy jsou až na jeden parametr totožné s neuronovou sítí typu Perceptron. Tímto parametrem se stává takzvaná šířka vah, nebo také velikost rozptylu. Tento parametru pak v architektuře RBF sítě napomáhá k překrývání jednotlivých rozptylů neuronů střední (skryté) vrstvy.

Vstupními parametry jsou si tedy oba typy sítí velice podobné. Velký rozdíl však nastává v algoritmizačním řešení. To samozřejmě vyplývá z rozdílných architektur sítí tak, jak jsou popsány v teoreticko-metodologické části práce.

Kromě podpůrných metod třídy je hlavní výpočtovou složkou třídy RBF metoda `vypocet_Neuron` a `vypocet_Odhad`. K těmto metodám se však přidává ještě metoda `nastaveni_vstupu`, která řeší algoritmický postup pro shlukovou analýzu a tvorbu jednotlivých ukazatelů a hodnot, které v dalších metodách přebírají neurony střední vrstvy.

Vzhledem k tomu, že u sítě typu RBF není potřebný back-propagation algoritmus vah mezi vstupními neurony a neurony střední (skryté) vrstvy, je chod nástroje pro simulaci neuronové sítě typu RBF časově méně náročný, nežli u nástroje simulace sítě typu Perceptron. Samozřejmě tento fakt platí, při zachování stejným parametrických nastavení. Celkově je programová struktura nástroje pro perceptronovou síť umístěna na zhruba čtyři sta třiceti řádcích kódu.

Celkový pohled na třídu RBF a její kompletní výpis metod a proměnných přináší následující obrázek 3.3.



Obrázek 3.3 Metody a proměnné tříd RBF

3.4 Výstup simulátoru pro chod neuronových sítí

Výstup simulátoru chodu neuronových sítí je víceméně podobný oběma typům simulovaných architektur. Dá se říci, že výstupem simulátoru je kompletní textový protokol, který obsahuje všechny důležité aspekty důležité pro další využití. Textový protokol je plně nastavitelný v rámci jednotlivých tříd simulátoru. Je tedy možné zobrazit průběh výpočtu od konečných odhadnutých výsledků až po nejmenší detaily v jednotlivých krocích, jako jsou například hodnoty potenciálů apod.

Základní nastavení výstupu simulátoru, tedy základní nastavení výstupního protokolu je následující:

- 1) Informace o zahájení chodu sítě
- 2) Výpis jednotlivých vstupních dat pro kontrolu správnosti
- 3) Hodnoty původních vah, výsledky shlukové analýzy u RBF sítě
- 4) V rámci chodu simulátoru postupný výpis čísla epochy, ve kterém se výpočet nachází.
- 5) V rámci druhé a poslední epochy kompletní výpis odhadu sítě na trénovací a validační množině.
- 6) Vypsáno pole chyb pro validační a trénovací množinu
- 7) Vypsán odhad sítě na trénovací a validační množině
- 8) Vypsána epocha, v níž dosáhla síť nejnižší hodnoty chyby

Toto je tedy základní nastavení, které lze bez větších problémů upravovat. Výstupní hodnoty jsou také připraveny pro další analýzu, jako je například grafické zobrazení hodnot ve vhodném softwaru apod. Kompletní protokol je tedy velmi obsáhlý, v plném nastavení pak nabývá desítek tisíc řádků textu, v závislosti na počtu kol učení. Jeho plné znění tak není možné uvést obrázkem ani přímým výpisem. Jeho ukázka je k dispozici na přiloženém kompaktním disku.

Jednotlivou aplikací tohoto simulátoru neuronových sítí na konkrétních vstupních datech a jeho výsledky se zabývá následující kapitola.

4 Ověření a porovnání predikční přesnosti simulátoru se statistickými modely

Pro ověření a porovnání přesnosti predikce časových řad z hlediska vytvořeného simulátoru neuronových sítí a statistickými ARIMA modely budou využity tři různé časové řady. Tyto časové řady budou řady vývoje cen akcií různých firem. Pro lepší přehlednost bude odstraněna časová složka naměřených dat a bude nahrazena hodnotou pozorování. V popisu časové řady však bude určen časový odstup mezi jednotlivými pozorováními. Každá časová řada použitá v této diplomové práci, je dostupná skrze webové stránky finance.yahoo.com.

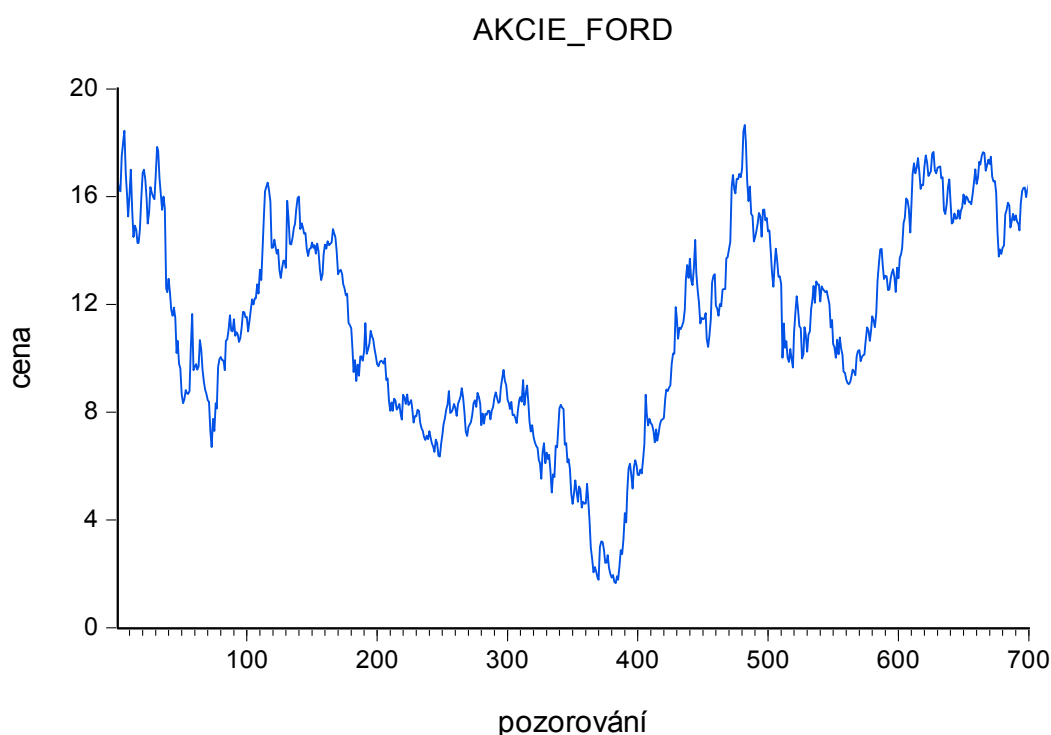
Každá časová řada podstoupí stejný postup analýzy a jejího následného využití pro porovnání predikční přesnosti. Postup je vždy následující:

- 1) Představení časové řady, určení rozdělení časové řady na validační a trénovací část a doplnění základních statistických hodnot
- 2) Využití ARIMA modelu nad danou časovou řadou, při využití postupů z teoreticko-metodologické části práce. Tedy určení modelu, jeho využití ve statické predikci a následná grafická a popisná část výsledků
- 3) Využití neuronových sítí typu RBF a Perceptron v rámci naprogramovaného simulátoru nad danou časovou řadou. Presentace vybraných dosažených výsledků pro jednotlivé konfigurace. Dále grafická a popisná prezentace výsledků pro výsledkově nejlepší konfiguraci jednotlivého typu sítě, včetně porovnání vývoje chyby v průběhu pozorování dané časové řady.

Použitým kritériem pro porovnání predikční přesnosti bude ukazatel RMSE pro validační množinu. Pro celkový postup vytváření ARIMA modelů a tvorbu grafů byl využit statistický software EViews.

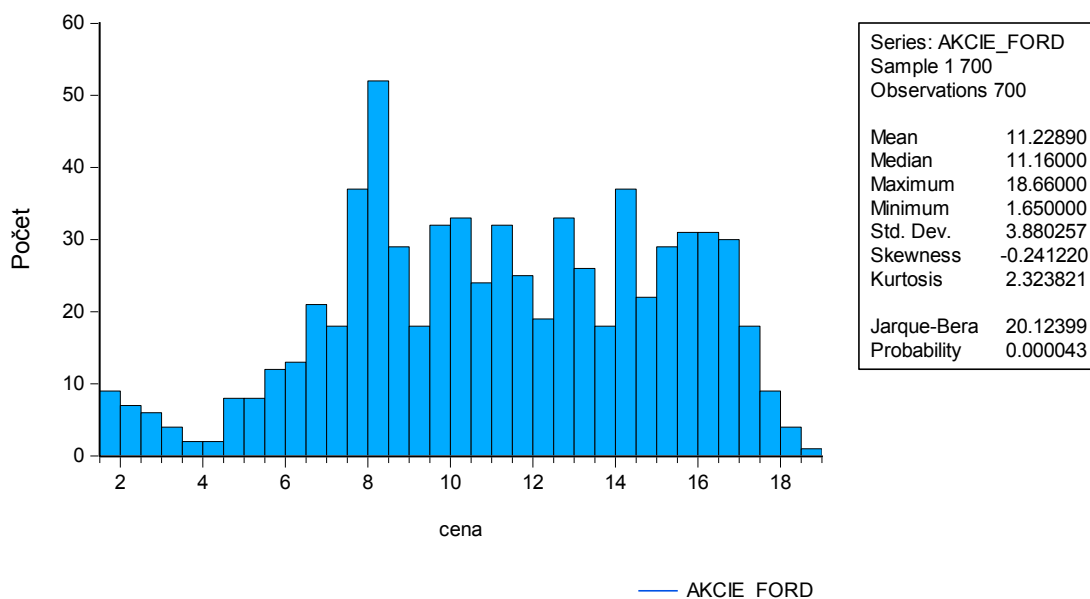
4.1 Akcie Ford Motor Co.

V prvním případě bude pro možnosti prognózování ekonometrickými nástroji a nástroji soft computingu použita časová řada nesoucí data o vývoji ceny akcií americké nadnárodní korporace Ford Motor Co., v amerických dolarech. Tato korporace se specializuje na výrobu automobilů. Naměřená data jsou v týdenní frekvenci. Cena uvedených akcií je cena otevírací v daný den. Celkem je k dispozici 700 pozorování. Data jsou tedy k dispozici o 29.10. roku 2001 do 23.3. roku 2015. Vzhledem k možnosti použít několik vstupních neuronů, je časová řada pro konečné prognózování upravená na 696 datových záznamů. Celková časová řada je rozdělená na trénovací a validační množinu v poměru zhruba 80 % ku 20 %. Trénovací množina tak má k dispozici 560 hodnot, validační množina hodnot 136. Vývoj celkové původní časové řady je znázorněn na obrázku 4.1.



Obrázek 4.1 Původní časová řada akcií Ford

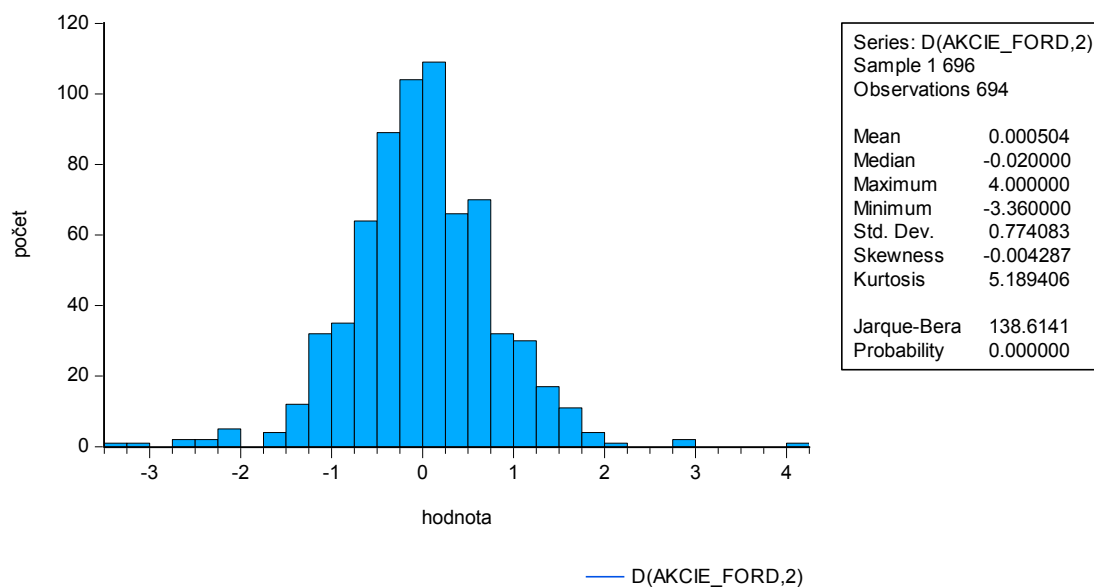
Základní charakteristiky původní časové řady pak doplní obrázek 4.2.



Obrázek 4.2 Histogram a statistika původní časové řady

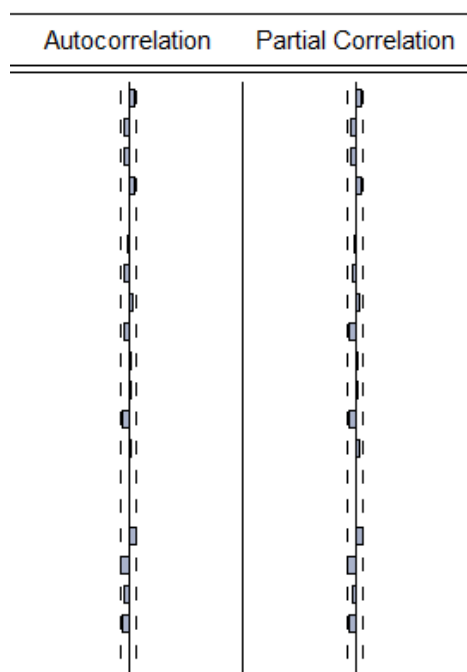
4.1.1 Modelování ARIMA

Již pohled na histogram původních hodnot vede k úvaze, že je daná časová řada nestacionární. Pohled na nestacionaritu časové řady poskytne unit root test, konkrétně ukazatele p-value, který vykazuje hodnotu 0,1902. Původní časová řada je tedy nestacionární. Nestacionaritu se pokusí odstranit diferencí. Po první diferenci již časová řada v unit root testu vykazuje hodnotu p-value 0,0000. Stacionarita časové řady je tedy zajištěna již po první diferenci a bude samozřejmě stacionární i pro další difference. Například histogram pro druhou diferenci je vidět na obrázku 4.3.

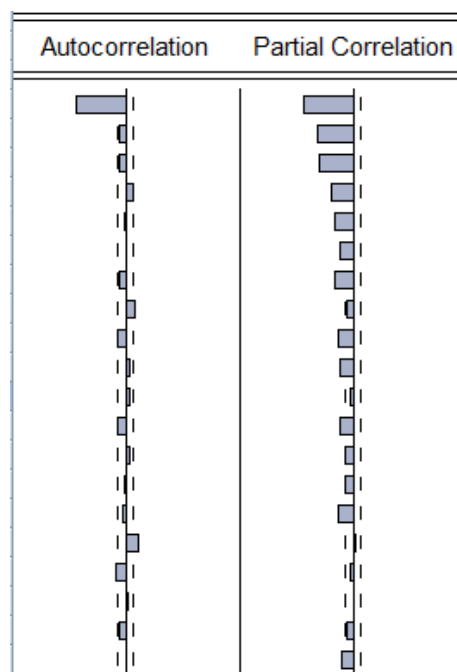


Obrázek 4.3 Histogram druhé difference

Identifikování modelu ARIMA pro tuto časovou řadu slouží grafy vývoje autokorelační a parciální autokorelační funkce. Na obrázku 4.4 jsou znázorněny ACF a PACF pro první diferenci, na obrázku 4.5 pak ACF a PACF pro druhou diferenci časové řady.



Obrázek 4.4 ACF a PACF první difference



Obrázek 4.5 ACF a PACF druhé difference

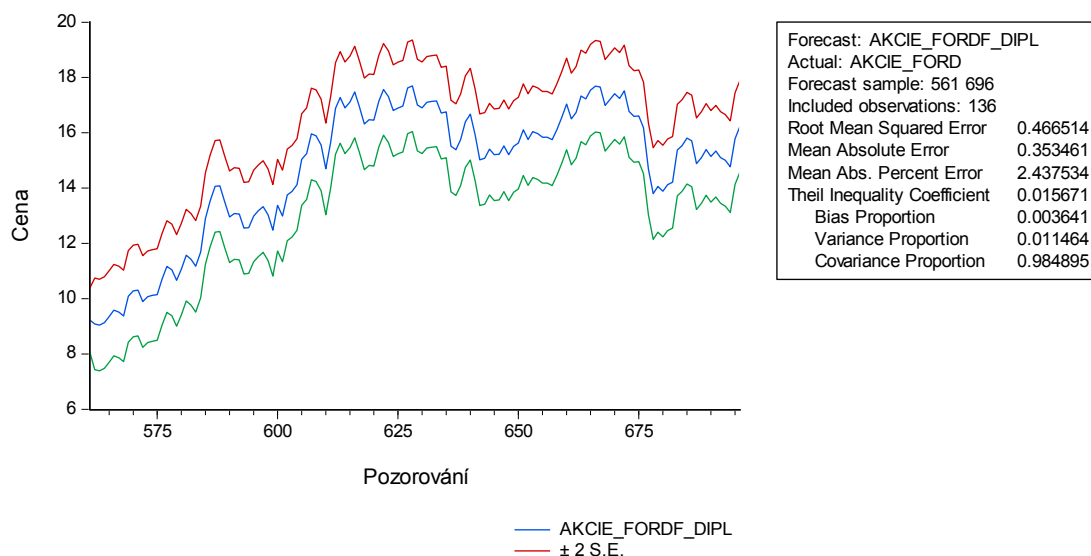
Pro první diferenci se po grafické analýze zdá vhodné použití modelu ARIMA(1,1,1). Nicméně tento model nedisponuje dobrými hodnotami vzhledem k významnosti koeficientů. Také R-square modelu je velmi nízké na hodnotě 0,001.

Model pro druhou diferenci ARIMA se jako nejvhodnější nabízí model ARIMA(0,2,1). Jeho koeficienty a statistika je uvedena na obrázku 4.6.

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	7.79E-05	0.000191	0.406745	0.6844
MA(1)	-0.996136	0.002731	-364.7384	0.0000
R-squared	0.478474	Mean dependent var	-1.79E-05	
Adjusted R-squared	0.477536	S.D. dependent var	0.811906	
S.E. of regression	0.586859	Akaike info criterion	1.775514	
Sum squared resid	191.4885	Schwarz criterion	1.791014	
Log likelihood	-493.3684	Hannan-Quinn criter.	1.781567	
F-statistic	510.1027	Durbin-Watson stat	1.924769	
Prob(F-statistic)	0.000000			

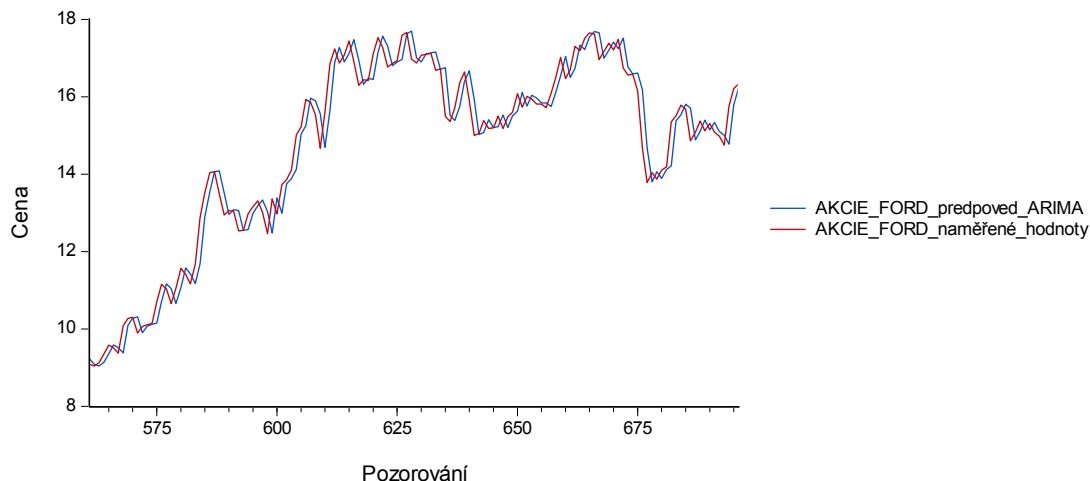
Obrázek 4.6 Koeficienty a statistika modelu ARIMA

Tento model je tedy možno využít pro statický odhad hodnot validační množiny. Statický odhad je uskutečněn nad validační množinou časové řady o celkovém počtu 136 pozorování. Samotný odhad ARIMA modelu je patrný na obrázku 4.7.



Obrázek 4.7 Odhad modelu ARIMA

Jak vyplývá z obrázku a její přiložené statistice je RMSE odhadu tohoto ARIMA modelu 0,466514. Pohled na validační část časové řady původních hodnot a hodnot odhadnutých ARIMA modelem představuje následující obrázek 4.8.



Obrázek 4.8 Porovnání predikce a původní validační množiny

Vzhledem k tomu že hodnoty trénovací množiny původní časové řady a hodnot ARIMA modelu jsou stejné, není potřeba srovnávat pro ARIMA modely řady trénovacích množin.

4.1.2 Modelování časové řady pomocí neuronových sítí

V rámci modelování časové řady neuronovými sítěmi je nutno vyzkoušet několik variant nastavení neuronových sítí. Tato nastavení se liší i v rámci jednotlivých typů neuronových sítí, vzhledem k jejich architektuře. V rámci odborné literatury neexistuje žádný ucelený soubor pravidel a postupů pro nastavování sítí. Je tedy nutno vyzkoušet mnoho různých kombinací tak, aby se síť dostala na minimum chybových funkcí.

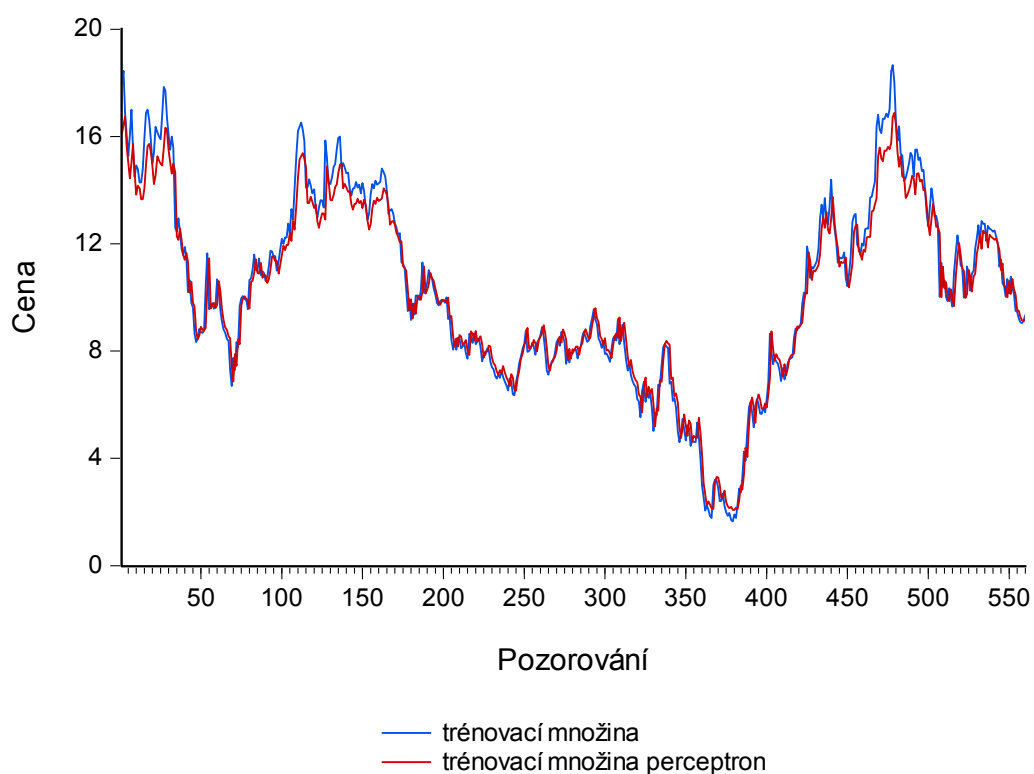
Perceptron

Neuronová síť typu Perceptron vykazuje po vyzkoušení různých kombinací nastavení různé parametry výstupu RMSE trénovací a validační množiny. Nejlepší vybrané hodnoty popisuje následující tabulka 4.1.

Tabulka 4.1 Parametry a výsledky sítě Peceptron

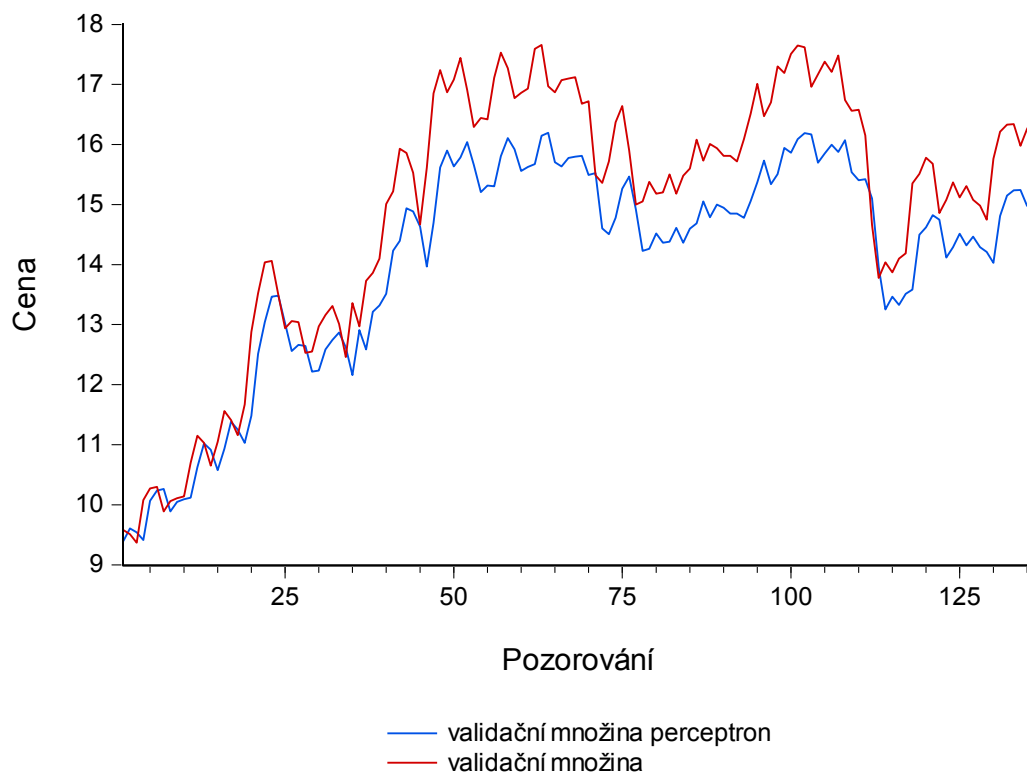
Počet vstupů	Počet kol učení	Rychlost učení	Počet neuronů střední vrstvy	RMSE trénovací	RMSE validační
1	50000	0,00001	50	0,69708	1,031529
1	50000	0,00001	10	0,707	1,1285247
1	50000	0,01	5	4,065	5,27815

Tabulka vykazuje pouze výběr vyzkoušených kombinací v rámci trénování neuronové sítě typu perceptron. Ač byly k dispozici až čtyři vstupní neurony, nejlepší výsledky vykazovaly síť s jedním vstupem. Nejlepší výsledky pak nese neuronová síť s padesáti neurony střední vrstvy o RMSE validační množiny 1,031529. Vývoj trénovací množiny této neuronové sítě vzhledem k vývoji trénovací množiny původní časové řady je znázorněn následujícím obrázkem 4.9.



Obrázek 4.9 Porovnání trénovací množiny sítě a původní trénovací množiny

Validační množinu neuronové sítě a validační množinu původní časové řady pak znázorňuje následující obrázek 4.10. Na samotných grafických výstupech je tedy patrné, že trénovací množina je daleko lépe zachycena neuronovou sítí, ačkoli má daleko větší poměr vstupních dat.



Obrázek 4.10 Validační množina sítě a původní validační množina

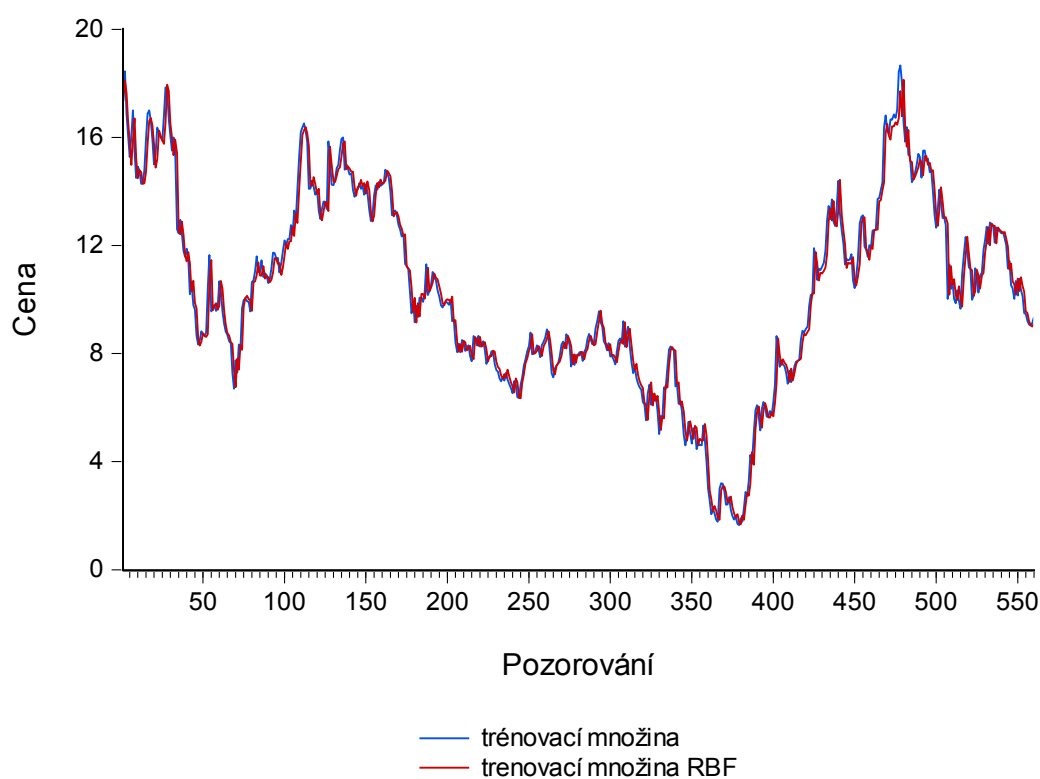
Neuronová síť typu RBF

Neuronová síť typu RBF vykazuje po vyzkoušení různých kombinací nastavení různé parametry výstupu RMSE. Oproti síti typu Perceptron je však u sítě typu RBF navíc parametr nastavení odchylky neuronů střední vrstvy, v této práci je značen jako šířka vah. Nejlepší vybrané hodnoty popisuje následující tabulka 4.2.

Tabulka 4.2 Parametry a výsledky sítě RBF

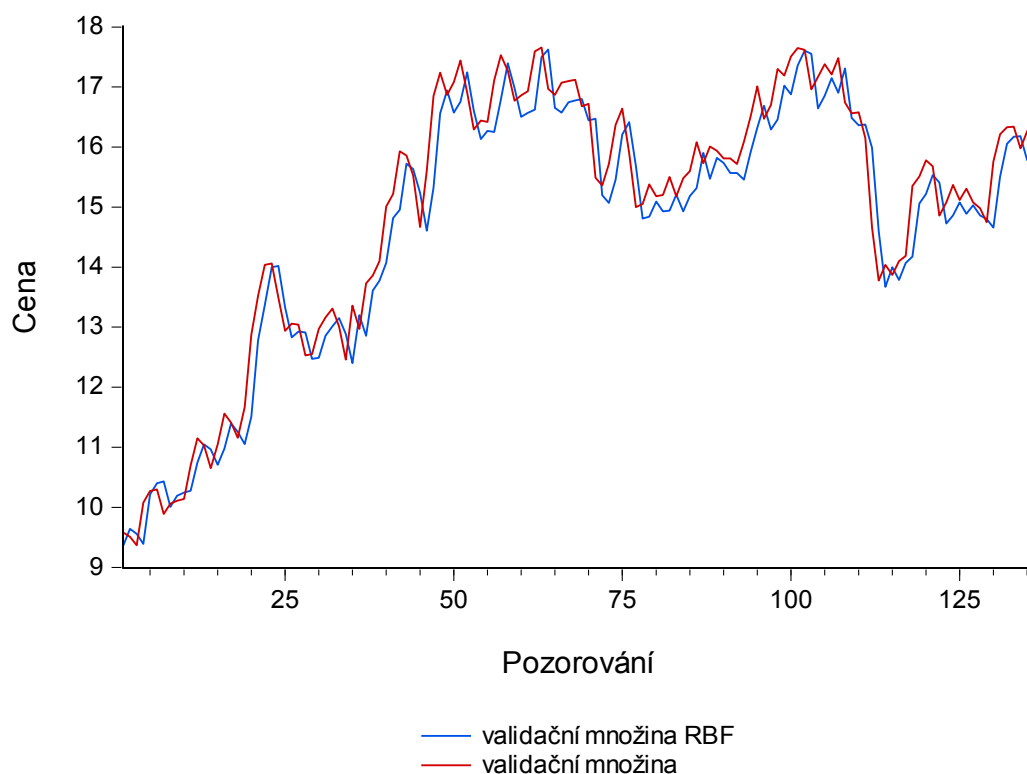
Počet vstupů	Počet kol učení	Rychlost učení	Počet neuronů střední vrstvy	Šířka vah	RMSE trénovací	RMSE validační
1	10000	0,001	150	10	0,58199	0,517822
1	10000	0,0001	50	10	0,586337	0,5201546
1	10000	0,01	5	100	1,002379	1,2001437

Jako u předchozí sítě, i u sítě typu RBF vykazují nejlepší výsledky nastavení sítě s jedním vstupním neuronem. Jako nejlepší byla vyhodnocena síť s parametry šířky vah 10 a počet neuronů střední vrstvy rovný 150. Jak je patrné z tabulky všechny RBF sítě byly trénovány na maximum 10 000 kol. To přináší velké časové úspory v porovnání s trénováním sítě typu perceptron. Vývoj trénovací množiny vybrané RBF sítě oproti trénovací množině původní časové řady uvádí následující obrázek 4.11.



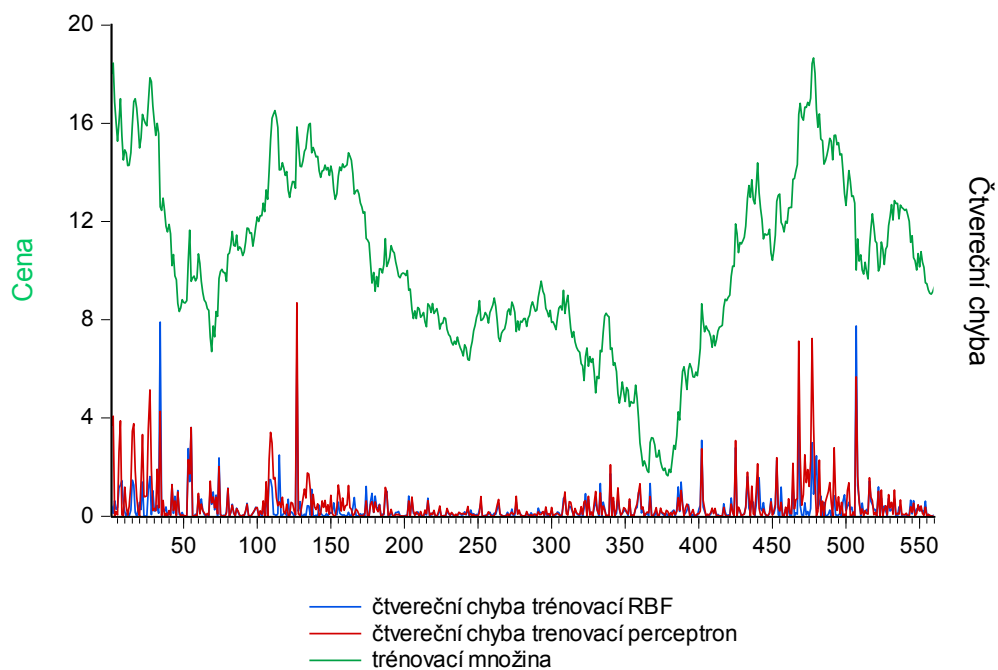
Obrázek 4.11 Porovnání trénovací množiny sítě a původní trénovací množiny

Vývoj validační množiny RBF sítě a validační množiny původní časové řady představuje následující obrázek 4.12.



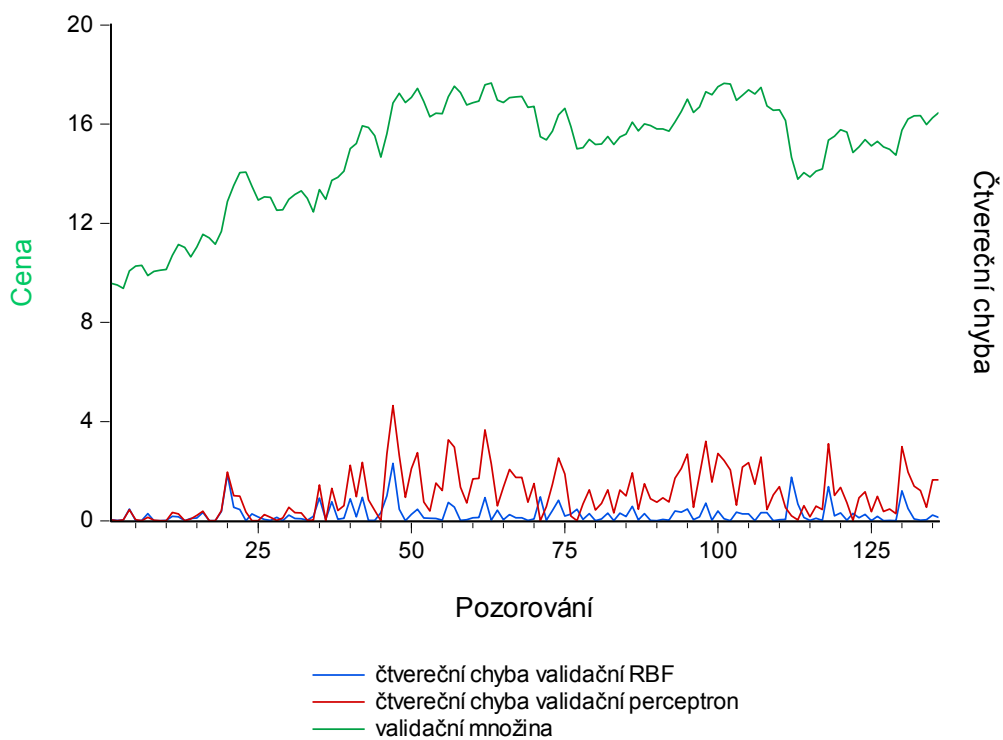
Obrázek 4.12 Porovnání validační množiny sítě a původní validační množiny

Jako velmi užitečné se ukazuje konečné srovnání vývoje čtvereční chyby v závislosti na jednotlivých pozorováních trénovací i validační množiny pro oba druhy neuronové sítě, doplněné o vývoj na daném druhu množiny původní časové řady. Srovnání vývoje čtvereční chyby trénovací množiny pro oba typy sítí ukazuje obrázek 4.13, doplněný o původní hodnoty trénovací množiny původní časové řady.



Obrázek 4.13 Vývoj chyby sítí v závislosti na vývoji trénovací množiny

Stejné srovnání, ale pro validační množinu časové řady přináší obrázek 4.14.



Obrázek 4.14 Vývoj chyby sítí v závislosti na vývoji validační množiny

Z těchto porovnání je patrné, že RBF síť je daleko méně náchylná k tvorbě chyby při výrazných náhlých poklesech hodnot datové řady v čase.

4.1.3 Konečné srovnání pro časovou řadu vývoje ceny akcií Ford

Jako hlavní kritérium pro srovnání slouží ukazatel RMSE pro validační množinu. Srovnání jednotlivých hodnot ukazatelů přináší tabulka 4.3.

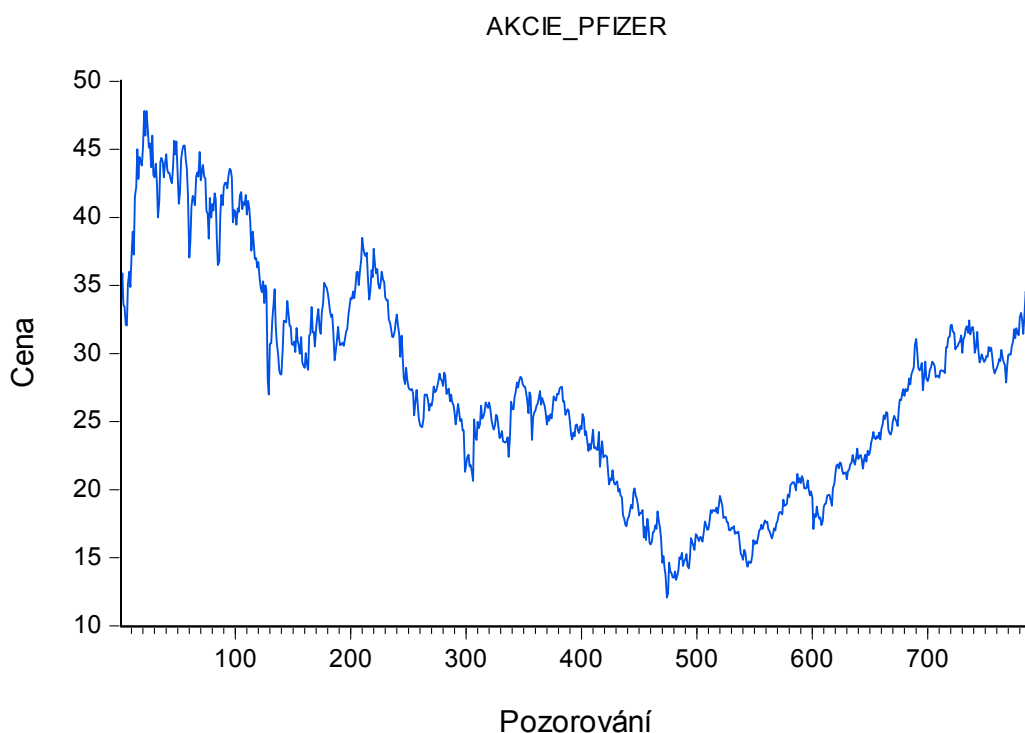
Tabulka 4.3 Porovnání výsledků RMSE

RMSE validační - Perceptron	1,031529
RMSE validační - RBF	0,517822
RMSE validační - ARIMA	0,466514

Je tedy patrné, že nejlepších výsledků statické prognózy přináší model ARIMA. Ačkoli síť typu RBF dosahuje velmi podobných výsledků. Síť typu Perceptron pak vykazuje zhruba dvojnásobné chyby statické prognózy validační množiny.

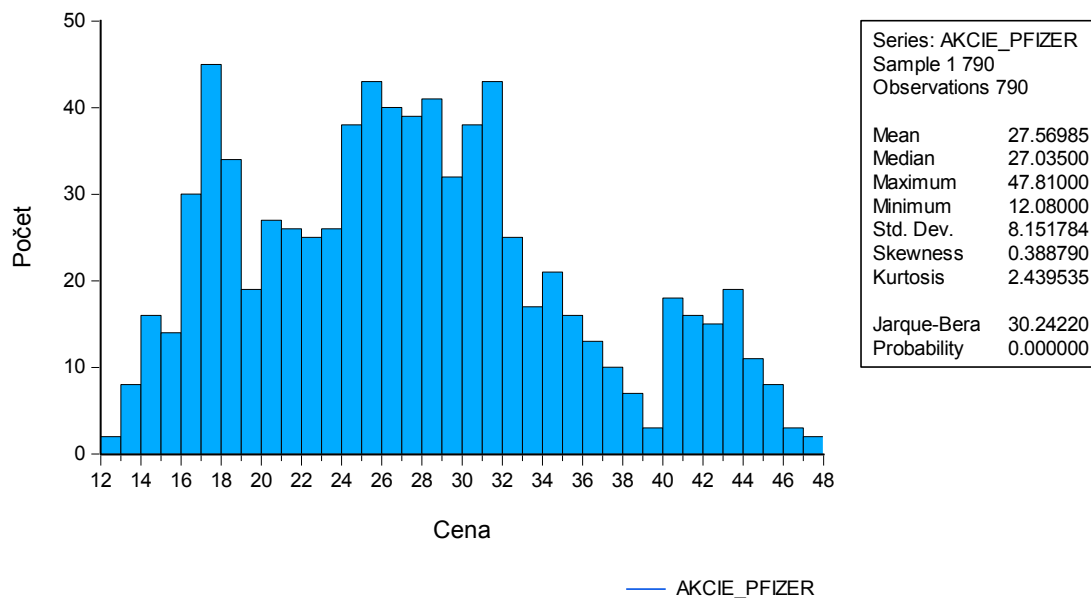
4.2 Akcie Pfizer Inc.

Druhou analyzovanou časovou řadou je časová řada vývoje ceny akcií společnosti Pfizer Inc. Tato společnost se pohybuje na trhu výroby léčiv. Časová řada obsahuje celkem 794 pozorování. Pro analýzu bude řada zkrácena o čtyři pozorování tak, aby byly k dispozici další vstupní řady pro neuronové sítě. Pro analýzu tak bude k dispozici 790 pozorování. Tentokrát bude časová řada rozdělena na trénovací a validační množinu v poměru 80 % ku 20 %. Trénovací množina tak obsahuje 632 pozorování, validační množina pak 158 pozorování. Naměřená data jsou týdenní, otevírací hodnoty akcií v daný den a jejich hodnota je uváděna v americkém dolaru. Pohled na celkovou časovou řadu ukazuje obrázek 4.15



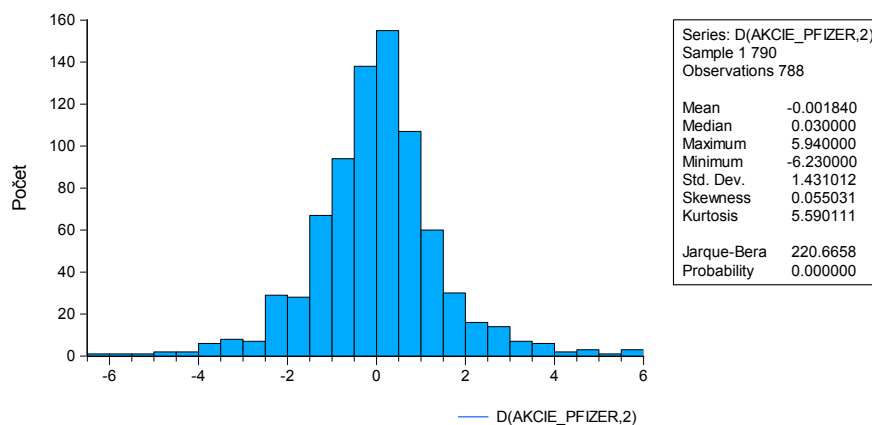
Obrázek 4.15 Vývoj původní časové řady

Základní statistické charakteristiky této časové řady doplňuje histogram z obrázku 4.16. Z tohoto histogramu je opět patrná nestacionarita časové řady, což potvrzuje také hodnota unit-root testu na hodnotě 0,54.



Obrázek 4.16 Histogram a statistika původní časové řady

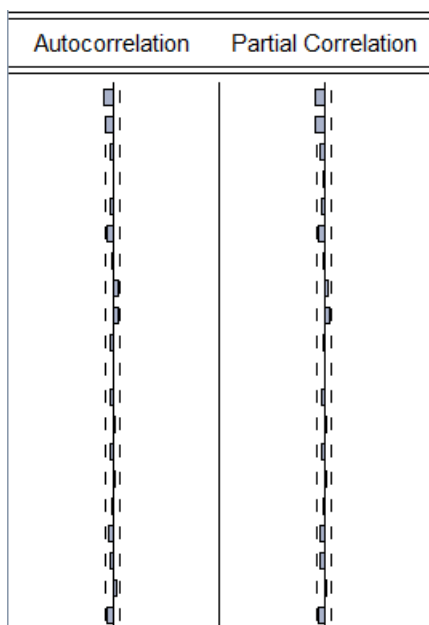
Odstranění nestacionarity spočívá opět v diferencování dané časové řady. Histogram první a druhé difference časové řady již ukazuje na stacionární časovou řadu, taktéž hodnota unit-root testu je v obou případech na nulové hodnotě. Časová řada je tedy po diferencování stacionární a připravená pro model ARIMA. Pohled na histogram druhé difference uvádí obrázek 4.17.



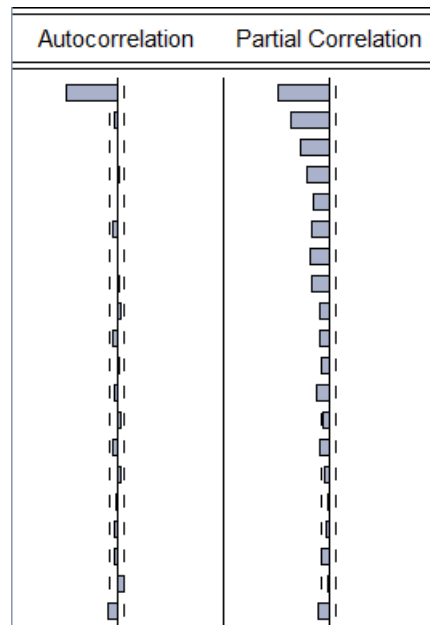
Obrázek 4.17 Histogram a statistika druhé difference

4.2.1 Modelování ARIMA

Pro identifikaci modelu ARIMA je opět využita autokorelační a parciální autokorelační funkce. Vývoje těchto funkcí jsou patrné na obrázcích 4.18 a 4.19, pro své první a druhé difference.



Obrázek 4.18 ACF a PACF první difference



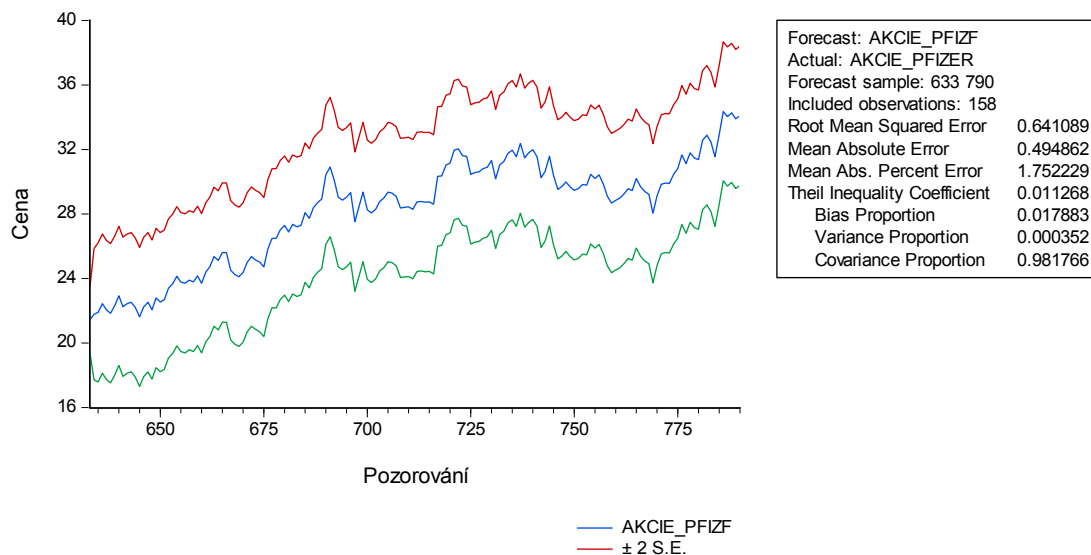
Obrázek 4.19 ACF a PACF druhé difference

Z těchto funkcí je patrné navržení modelu ARIMA(1,1,1) a ARIMA(1,2,2). Po navržení těchto modelů v příslušném statistickém softwaru, se jako vhodnější model jeví model ARIMA(1,2,2), který má nejvyšší hodnoty R-square a AIC. Pohled na kompletní statistiku modelu ARIMA(1,2,2) přináší následující obrázek 4.20.

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-2.49E-06	0.000231	-0.010804	0.9914
AR(1)	0.604222	0.181597	3.327270	0.0009
MA(1)	-1.699958	0.161627	-10.51777	0.0000
MA(2)	0.701308	0.161285	4.348256	0.0000
R-squared	0.542921	Mean dependent var		0.003863
Adjusted R-squared	0.540727	S.D. dependent var		1.520193
S.E. of regression	1.030230	Akaike info criterion		2.903780
Sum squared resid	663.3586	Schwarz criterion		2.932042
Log likelihood	-909.2388	Hannan-Quinn criter.		2.914758
F-statistic	247.4594	Durbin-Watson stat		1.980642
Prob(F-statistic)	0.000000			

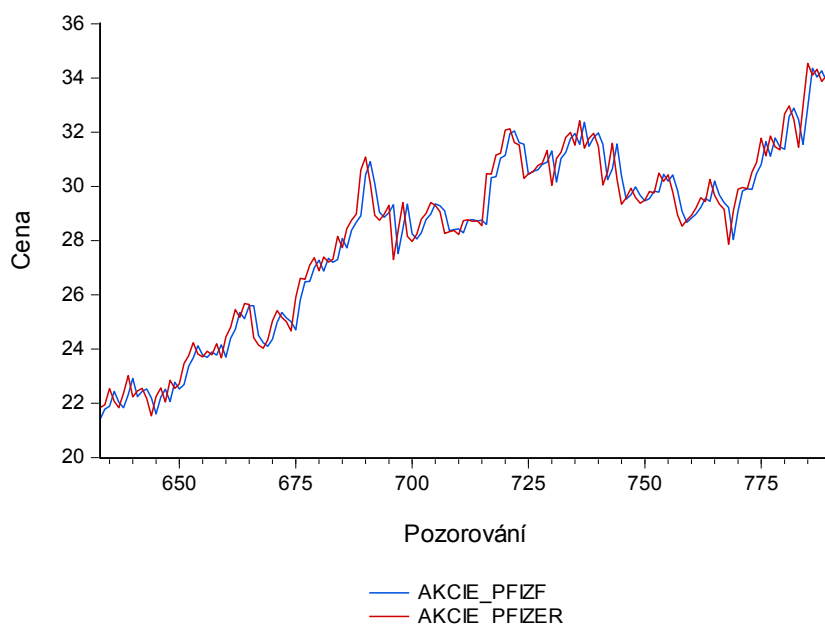
Obrázek 4.20 Statistika a koeficient modelu

Model ARIMA(1,2,2) je tedy využit pro statický odhad hodnot validační množiny. Validační množina obsahuje 158 pozorování. Celková statistika odhadu a pohled na vygenerovanou časovou řadu uvádí obrázek 4.21.



Obrázek 4.21 Odhad modelu

Jak je patrné, statický odhad modelu vykazuje chybu RMSE na hodnotě 0,641089. Přímé grafické porovnání původní časové řady validační množiny a statického odhadu ARIMA modelu zobrazuje následující obrázek 4.22.



Obrázek 4.22 Odhad modelu ve srovnání s původní validační množinou

4.2.2 Modelování časové řady pomocí neuronových sítí

Perceptron

Neuronová síť typu Perceptron vykazuje po natrénování následující hodnoty RMSE trénovací a validační množiny. V tabulce 4.4 byly opět použity tři nejlepší sítě, podle hodnot RMSE.

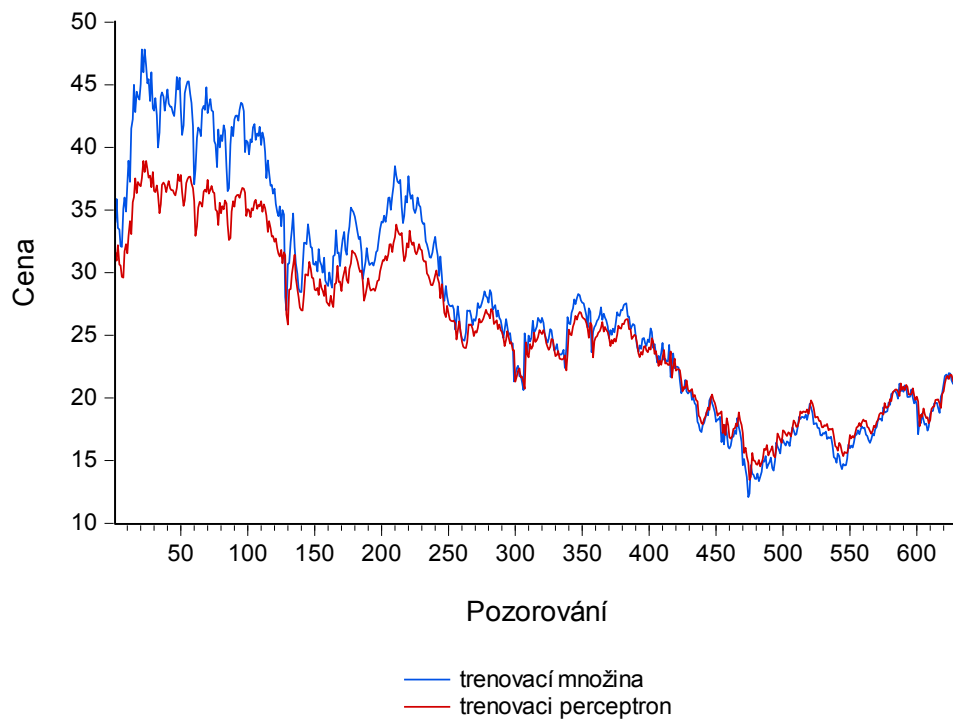
Tabulka 4.4 Parametry a výsledky sítě Perceptron

Počet vstupů	Počet kol učení	Rychlost učení	Počet neuronů střední vrstvy	RMSE trénovací	RMSE validační
1	200000	0,000001	70	1,1979	1,8262
1	200000	0,000001	40	1,203821	1,9578
1	200000	0,00001	50	1,2319	1,98224

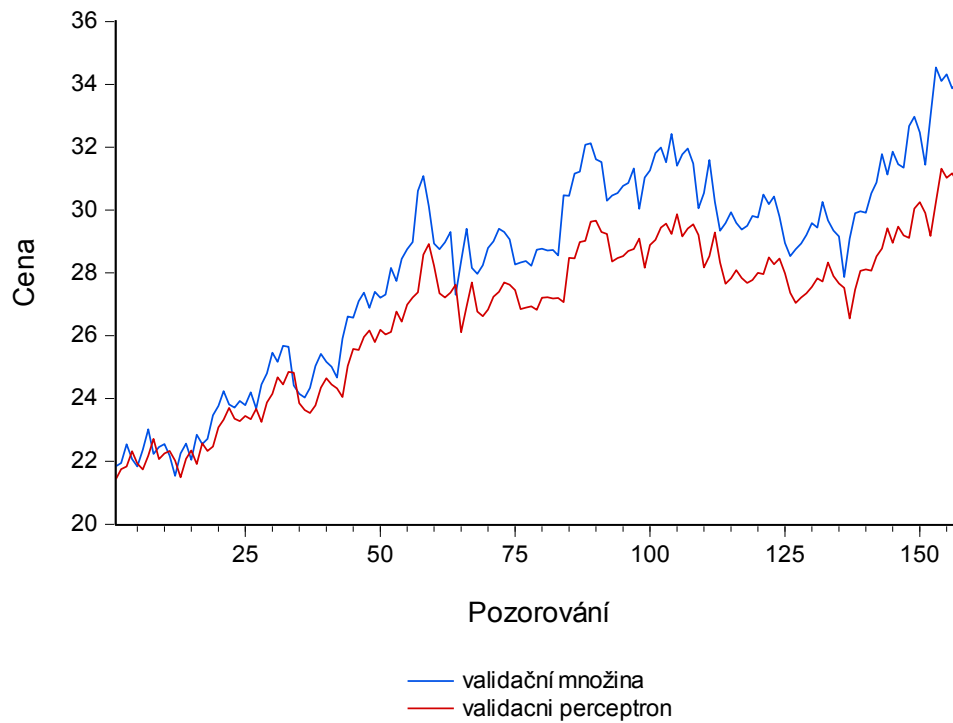
Trénování sítí typu Perceptron s tímto nastavením pro každou jednotlivou konfiguraci zabíralo více jak hodinu výpočetního času. K dispozici bylo opět více vstupních neuronů, nicméně nejlepší výsledky podávaly nastavení sítě s jedním vstupním neuronem. Počet kol učení sítě byl tentokrát nastaven až na hodnotu dvou set tisíc kol. Nejnižší hodnota RMSE validační množiny, na kterou se síť dokázala natrénovat je 1,8262. Síť s tímto nastavením je pak uváděna v následujících grafech pro zobrazení natrénovaných a odhadnutých hodnot.

Obrázek 4.23 pak ukazuje výsledky natrénovaných hodnot neuronové sítě typu Perceptron a původních hodnot trénovací množiny dané časové řady na příslušném grafu. Z grafu je pak patrné, že síť se nedokázala natrénovat zejména na prvních dvě stě pozorováních.

Podobné srovnání pro validační množinu časové řady uvádí obrázek 4.24. Je patrné, že síť odhaduje průběh validační množiny výrazně pod skutečnou hodnotou validační množiny.



Obrázek 4.23 Porovnání trénovací množiny sítě s původní trénovací množinou



Obrázek 4.24 Srovnání validační množiny sítě a původní validační množiny

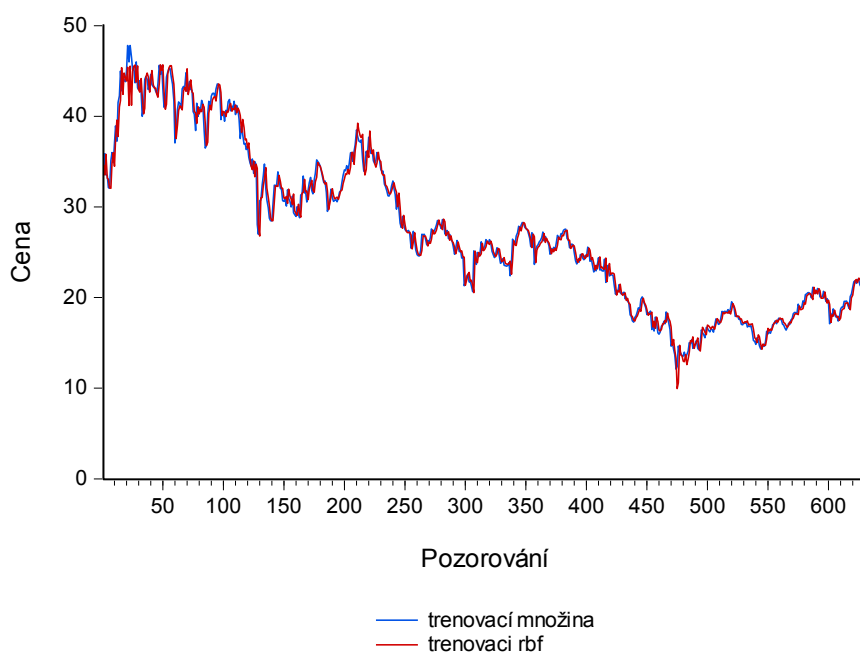
Neuronová síť typu RBF

Tabulka 4.5 ukazuje nejlepší tři kombinace nastavení parametrů RBF sítě z hlediska dosažení nejnižšího ukazatele RMSE na validační množině. Zcela nejnižší hodnoty RMSE validační množiny pak vykazuje síť s dvěma sty neurony skryté vrstvy. Tato síť a její výsledky budou následně graficky analyzovány.

Tabulka 4.5 Parametry a výsledky sítě RBF

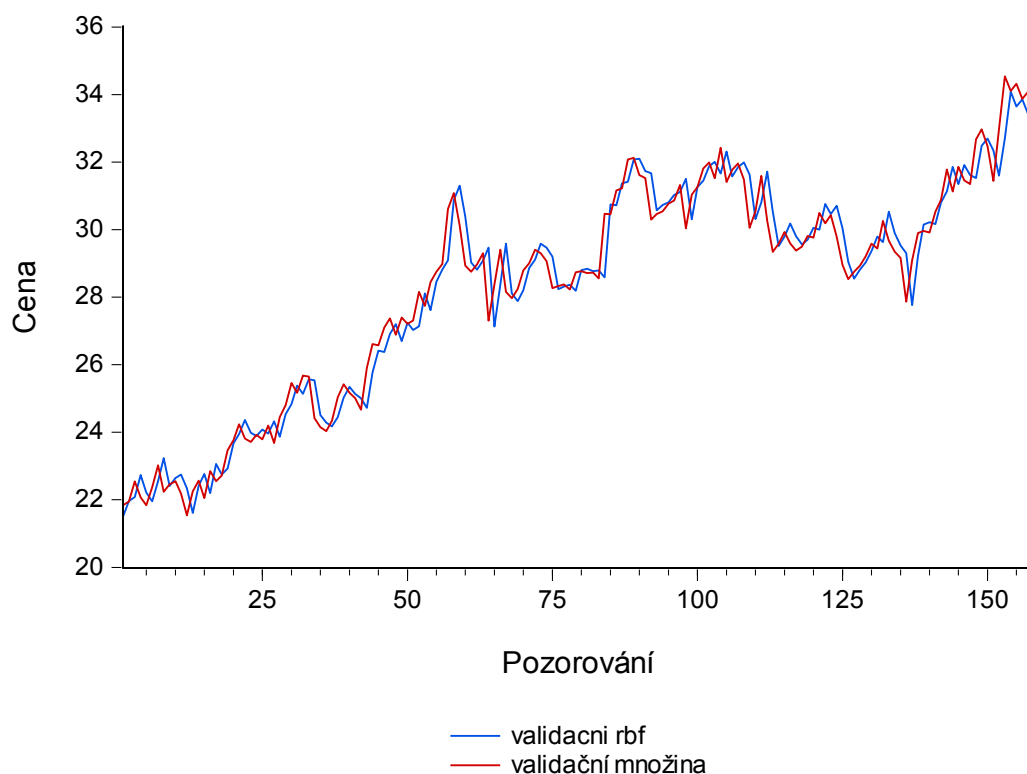
Počet vstupů	Počet kol učení	Rychlost učení	Počet neuronů střední vrstvy	Šířka vah	RMSE trénovací	RMSE validační
1	10000	0,0001	200	25	1,201	0,67157
1	80000	0,00001	500	40	1,0603	0,70143
1	80000	0,00001	600	100	1,0303	0,69073

Obrázek 4.25 vykazuje trénovací hodnoty vybrané RBF sítě v porovnání s původní trénovací množinou. Je patrné, že v této situaci síť RBF vykazuje daleko lepší výsledky v porovnání se sítí Perceptron. Trénovací množinu pak odhaduje velmi dobře.



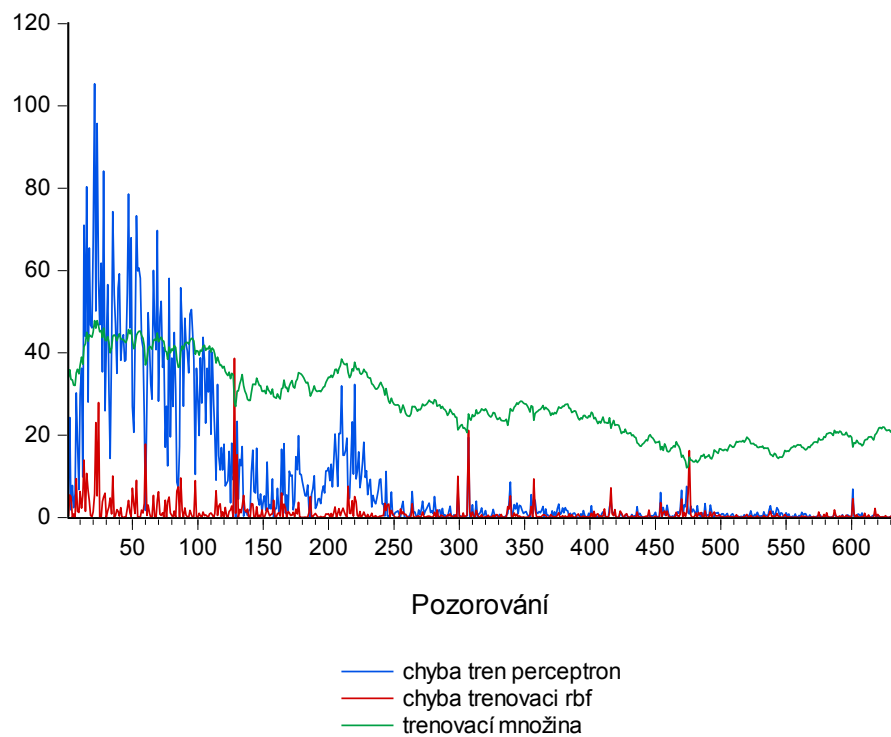
Obrázek 4.25 Srovnání trénovací množiny sítě a původní trénovací množiny

Pro porovnání výsledků odhadu validační části RBF sítě a původní validační množiny slouží obrázek 4.26. Výsledky jsou velmi dobré, srovnatelné s odhadem ARIMA modelu.

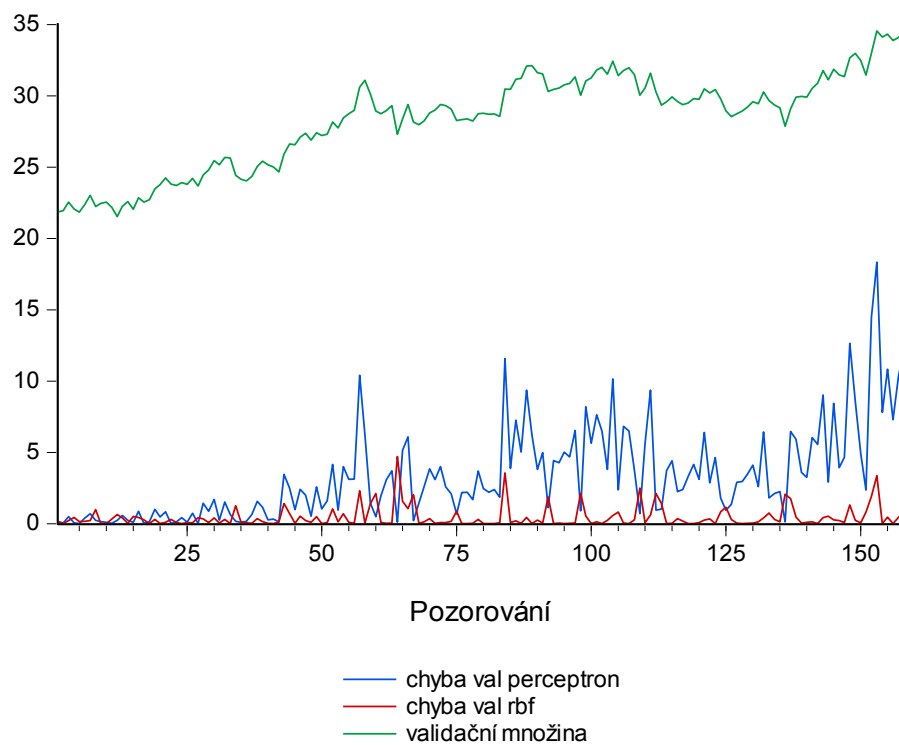


Obrázek 4.26 Srovnání validační množiny sítě a původní validační množiny

Srovnání vývoje čtverečních chyb jednotlivého typu neuronové sítě za jednotlivý pozorování, doplněné o původní trénovací a validační množinu uvádí následující obrázky 4.27 a 4.28. Z těchto srovnání je patrné perceptronová síť není již na trénovací množině schopna dosáhnout nízké chyby v jednotlivých prvotních pozorováních.



Obrázek 4.27 Vývoje chyb sítí v závislosti na vývoji trénovací množiny



Obrázek 4.28 Vývoje chyb sítí v závislosti na vývoji validační množiny

4.2.3 Konečné srovnání pro časovou řadu vývoje ceny akcií Pfizer

Jako hlavní kritérium pro srovnání slouží ukazatel RMSE pro validační množinu. Srovnání jednotlivých hodnot ukazatelů přináší tabulka 4.6.

Tabulka 4.6 Porovnání výsledků RMSE

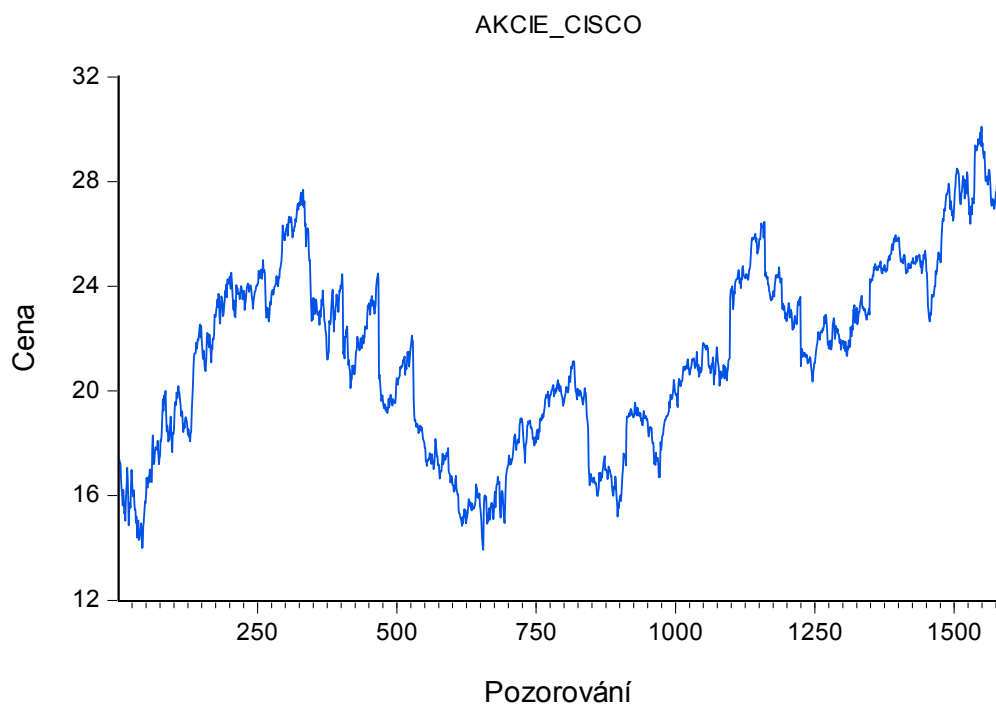
RMSE validační - Perceptron	1,8262
RMSE validační - RBF	0,67157
RMSE validační - ARIMA	0,641089

Nejlepší hodnoty RMSE po validační část, tedy statický odhad hodnot přináší ARIMA model. RBF síť však přináší pouze o zhruba tři setiny horší výsledek. Síť typu perceptron se na těchto datech neosvědčila a vykazuje vysokou hodnotu RMSE na hodnotě 1,8262.

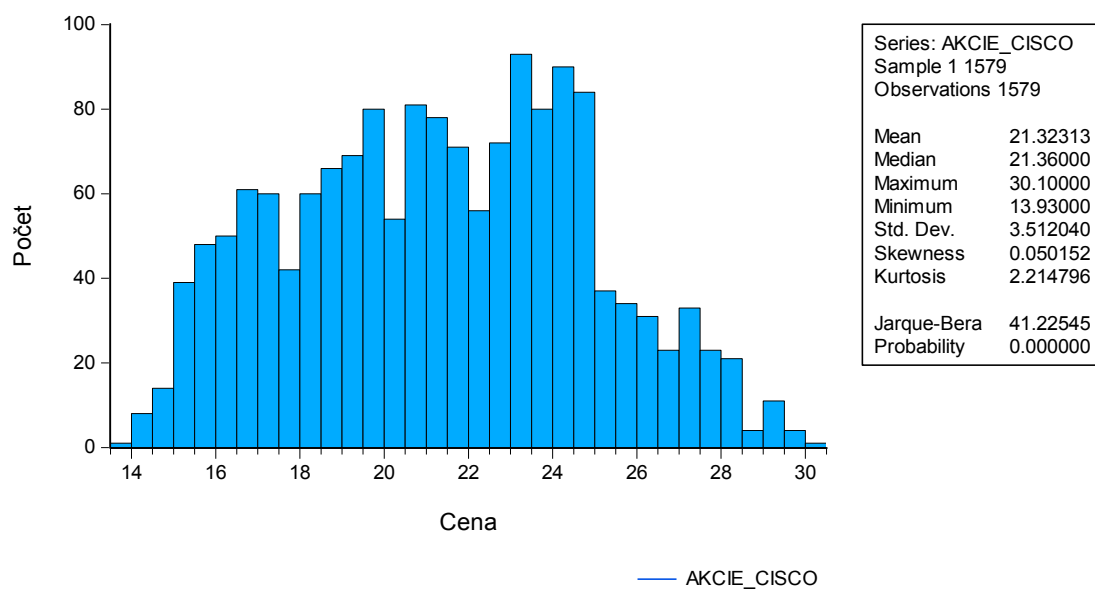
4.3 Akcie Cisco Systems Inc.

Poslední časová řada, která poslouží pro možnosti prognózování dat pomocí ekonometrického modelu a neuronových sítí, je časová řada vývoje akcií společnosti Cisco Systems Inc. Tato společnost je jednou z největších počítačových firem na světovém trhu. Specializuje se zejména na výrobu síťových prvků. Analyzovaná časová řada akcií společnosti Cisco Systems vykazuje celkově 1579 pozorování. Hodnotami časové řady jsou ceny akcií společnost v amerických dolarech. Jedná se o denní data, otevírací hodnoty akcií v daný den pozorování. Pro potřeby ARIMA modelu a neuronových sítí byla časová řada rozdělena na trénovací a validační část. Trénovací část vykazuje celkem první 1200 pozorování. Validační část obsahuje celkem 379 pozorování. Procentuální rozdělení je tedy 76 % pro trénovací a 24 % pro validační část.

Pohled na vývoj ceny akcií společnosti v závislosti na pozorováních přináší obrázek 4.29. Základní statistiku časové řady pak přináší histogram z obrázku 4.30.

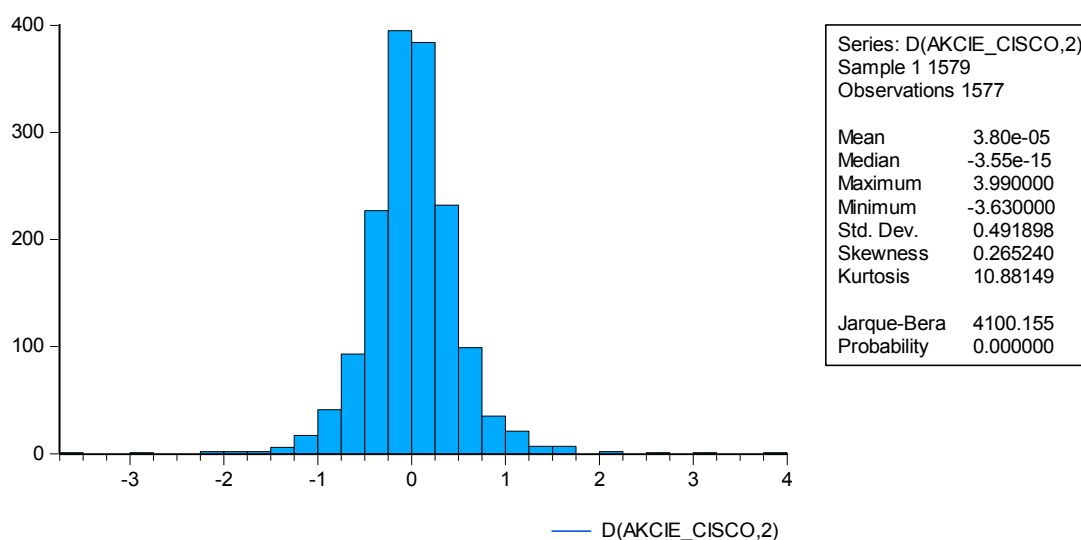


Obrázek 4.29 Vývoj původní časové řady



Obrázek 4.30 Histogram a statistika původní časové řady

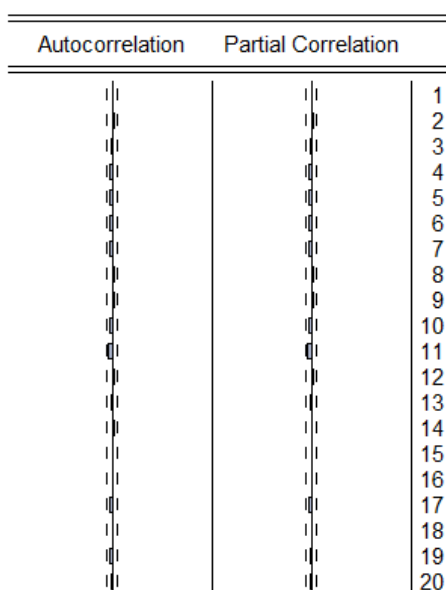
Jak je již patrné z histogramu, časová řada vykazuje nestacionaritu. Pro odstranění nestacionarity byla použita difference časové řady a již od první difference vykazuje stacionární chování, což podporuje také unit-root test, který je na nulové hodnotě již od použité první difference. Jak je vidět na obrázku 4.31 druhé difference a jejího histogramu, je časová řada stacionární a vykazuje nulovou střední hodnotu.



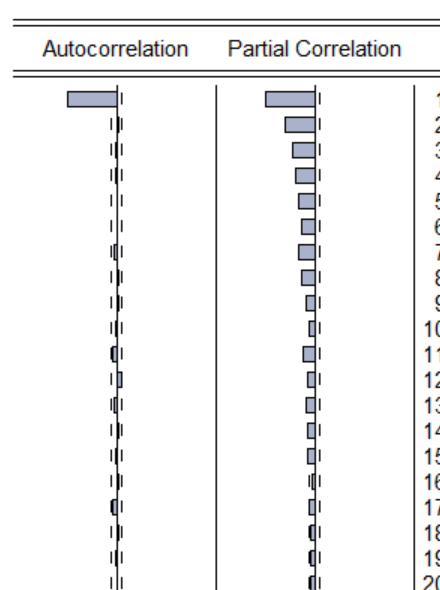
Obrázek 4.31 Histogram a statistika druhé difference

4.3.1 Modelování ARIMA

Pro identifikaci modelu byly opět využity ACF a PACF první a druhé difference časové řady jak je vidět na obrázku 4.32 a 4.33.



Obrázek 4.32 ACF a PACF první difference



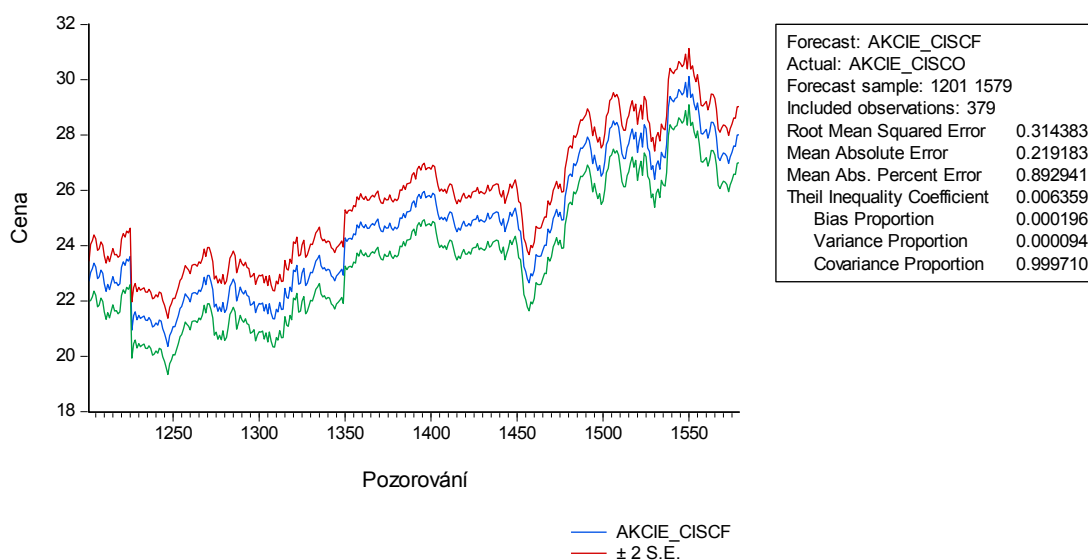
Obrázek 4.33 ACF a PACF druhé difference

Z výše uvedené analýzy je možné využití modelu ARIMA(1,2,1) nebo ARIMA(0,2,1). Modely první difference nesplňují kritéria použití modelu pro nevýznamnost koeficientů. Použitelným modelem v rámci kritéria AIC a významnosti koeficientů se ukázal být model ARIMA(0,2,1). Charakteristiky tohoto modelu, včetně konstanty a koeficientu MA(1) jsou patrné z obrázku 4.35.

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	2.92E-06	3.17E-05	0.092049	0.9267
MA(1)	-0.998945	0.001237	-807.6132	0.0000
R-squared	0.485116	Mean dependent var	2.97E-18	
Adjusted R-squared	0.484686	S.D. dependent var	0.501596	
S.E. of regression	0.360073	Akaike info criterion	0.796646	
Sum squared resid	155.0642	Schwarz criterion	0.805141	
Log likelihood	-475.1912	Hannan-Quinn criter.	0.799847	
F-statistic	1126.855	Durbin-Watson stat	1.944027	
Prob(F-statistic)	0.000000			

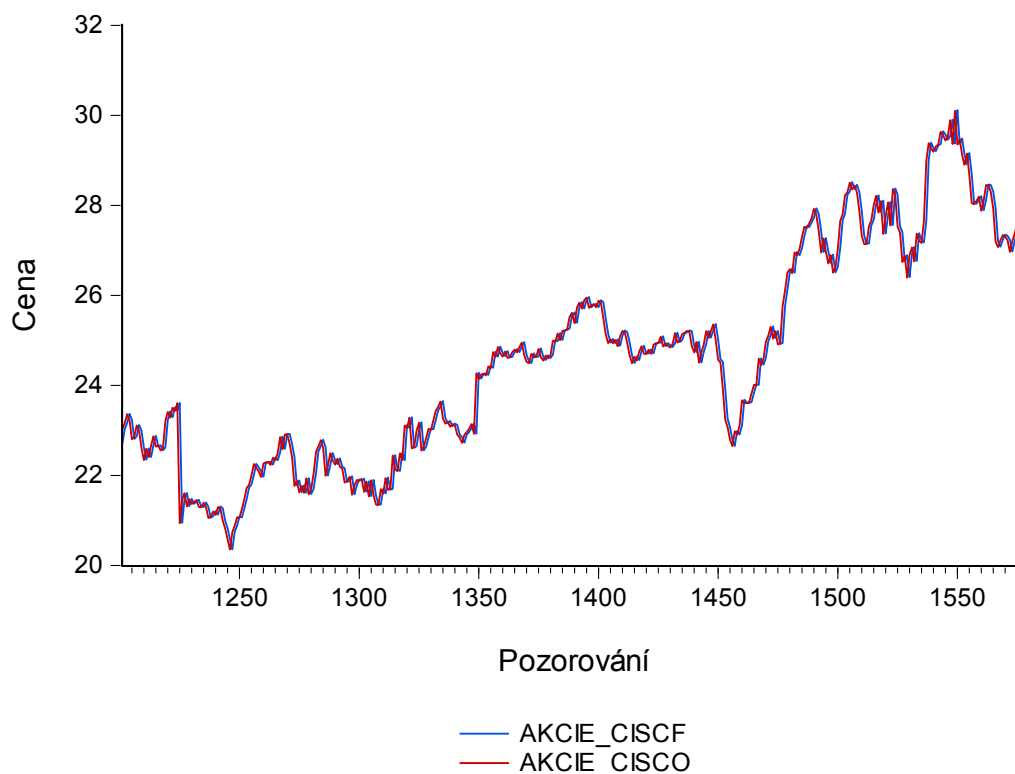
Obrázek 4.34 Statistika a koeficienty modelu

Tento model byl tedy využití pro statický odhad hodnot nad validační množinou časové řady. Výsledky tohoto odhadu jsou patrné z obrázku 4.35. Výsledná hodnota RMSE je pro tento odhad 0,31483.



Obrázek 4.35 Odhad modelu

Srovnání výsledného odhadu ARIMA modelu nad validační množinou a původní validační množiny přináší obrázek 4.36.



Obrázek 4.36 Odhad modelu ve srovnání s původní validační množinou

4.3.2 Modelování časové řady pomocí neuronových sítí

Perceptron

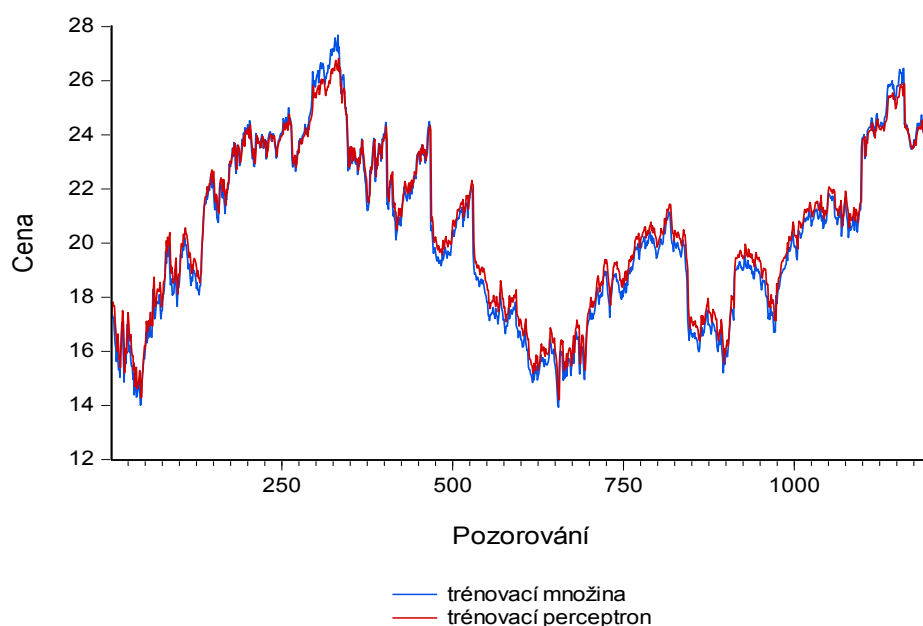
Neuronová síť typu Perceptron vykazuje po natrénování následující hodnoty RMSE trénovací a validační množiny. V tabulce 4.7 byly opět použity tři nejlepší sítě, podle hodnot RMSE.

Tabulka 4.7 Parametry a výsledky sítě Perceptron

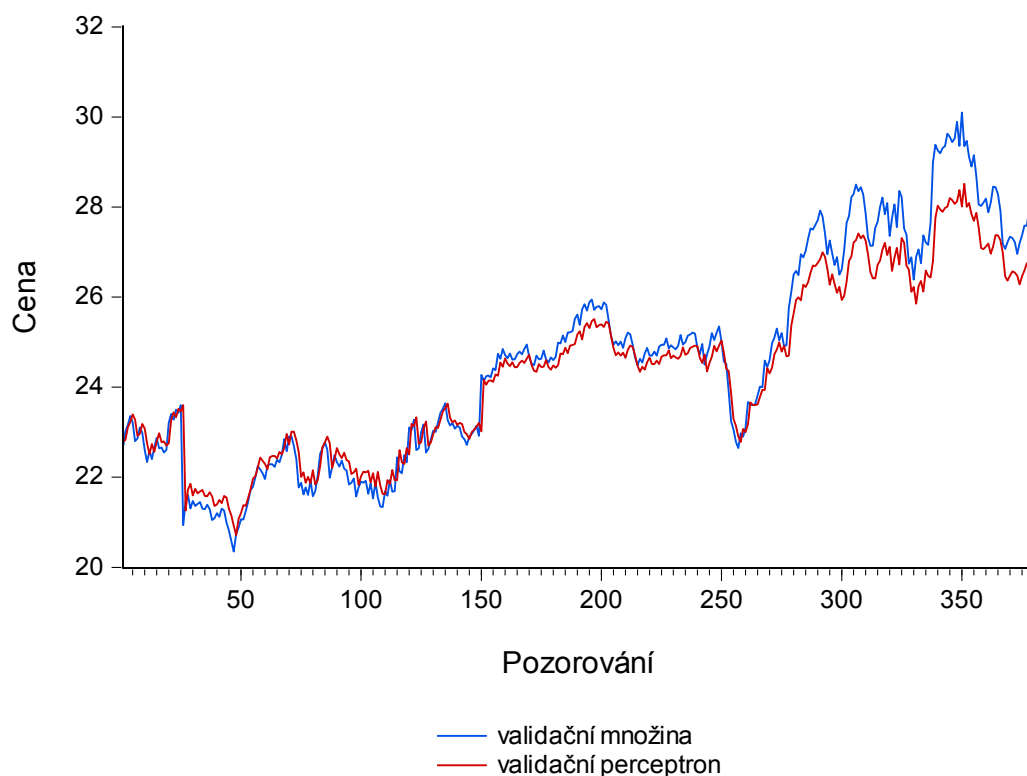
Počet vstupů	Počet kol učení	Rychlost učení	Počet neuronů střední vrstvy	RMSE trénovací	RMSE validační
1	80000	0,00001	120	0,13765	0,603
1	80000	0,00001	80	0,14018	0,684
1	80000	0,00001	50	0,38721	0,709

Trénování sítě typu Perceptron vykazovalo nejlepší hodnoty RMSE právě pro síť s jedním vstupem. Trénování bylo opět časově velmi náročné. To bylo zapříčiněno zejména využitím trénovací množiny o větším počtu pozorování, nežli u předchozích časových řad. Nejlepší výsledek RMSE validační množiny představuje síť se sto dvaceti neurony skryté vrstvy na hodnotě 0,603. Tato síť a její hodnoty trénovací a validační množiny pak vstupuje do následující grafické analýzy.

Obrázek 4.37 pak ukazuje srovnání trénovací množiny neuronové sítě typu Perceptron a původní trénovací množiny. Stejně srovnání, ale pro validační množinu přináší následující obrázek 4.38.



Obrázek 4.37 Srovnání trénovací množiny sítě a původní trénovací množiny



Obrázek 4.38 Srovnání validační množiny sítě a původní validační množiny

Neuronová síť typu RBF

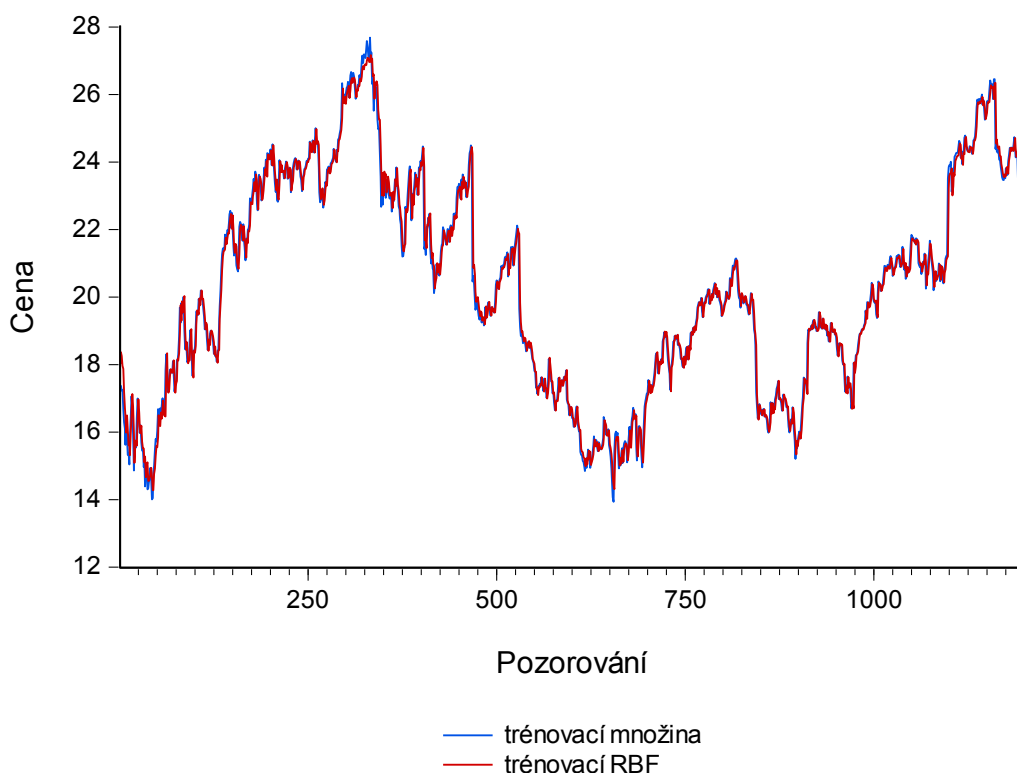
Tabulka 4.8 ukazuje vybrané tři kombinace nastavení parametrů RBF sítě z hlediska dosažení nejnižšího ukazatele RMSE na validační množině. Zcela nejnižší hodnoty RMSE validační množiny pak vykazuje síť s dvěma sty neurony skryté vrstvy. A to na úrovni 0,9711. Tato síť a její výsledky budou následně graficky analyzovány.

Tabulka 4.8 Parametry a výsledky sítě RBF

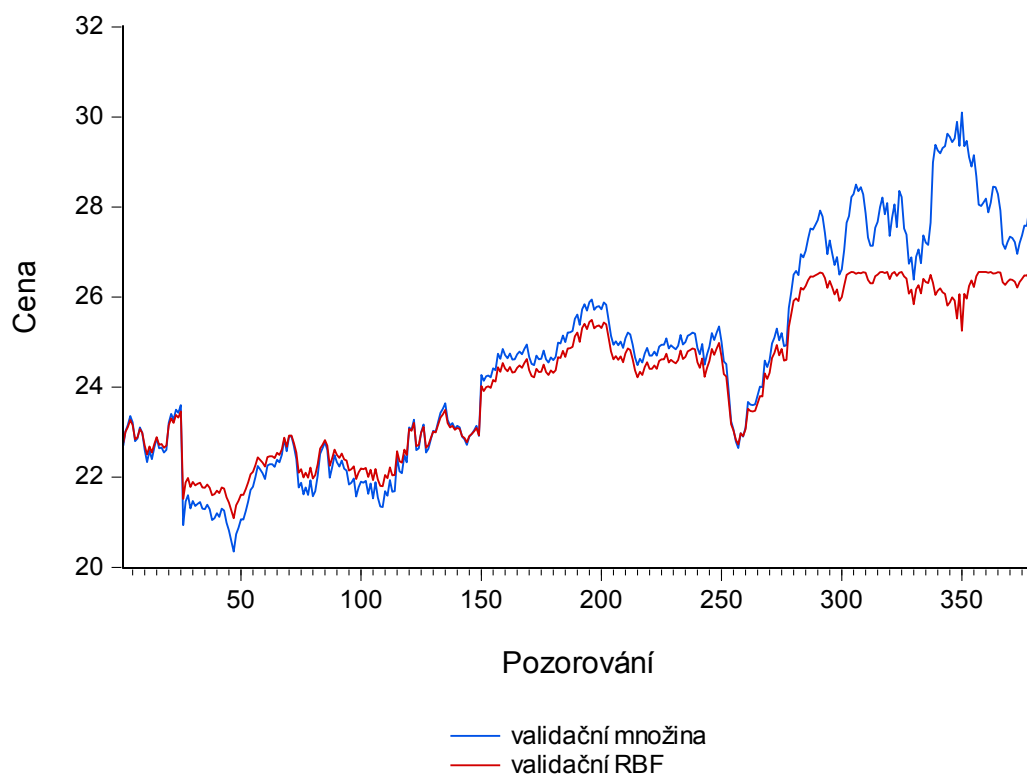
Počet vstupů	Počet kol učení	Rychlost učení	Počet neuronů střední vrstvy	Šířka vah	RMSE trénovací	RMSE validační
1	80000	0,001	200	200	0,1377	0,9711
1	80000	0,0001	120	200	0,1379	1,3696
1	80000	0,00001	120	5	0,1774	7,7552

Zajímavá je však ukázka rozdílu v RMSE validační množiny u sítě s dvěma sty a sto dvaceti neurony ve skryté vrstvě. Ačkoli mají tyto sítě téměř stejný výsledek RMSE trénovací množiny, na validační množině již vykazuje síť se sto dvaceti neurony ve skryté vrstvě větší chybu na validační množině. Taktéž poslední síť nevykazovala výrazně horší výsledek na trénovací množině, avšak na množině validační není schopna správného odhadu a její chyba na validační množině je v rámci dat velmi vysoká. Na hodnotě 7,7552.

Obrázek 4.39 přináší srovnání trénovací množiny neuronové sítě a trénovací množiny původní časové řady. Obrázek 4.40 pak srovnání validační množiny neuronové sítě a validační množiny původní časové řady. Na srovnání validačních množin je patrné, že chyba odhadu velmi stoupne zhruba od dvou set šedesátého pozorování na této validační množině. Síť již na těchto hodnotách není schopná dobrého odhadu.

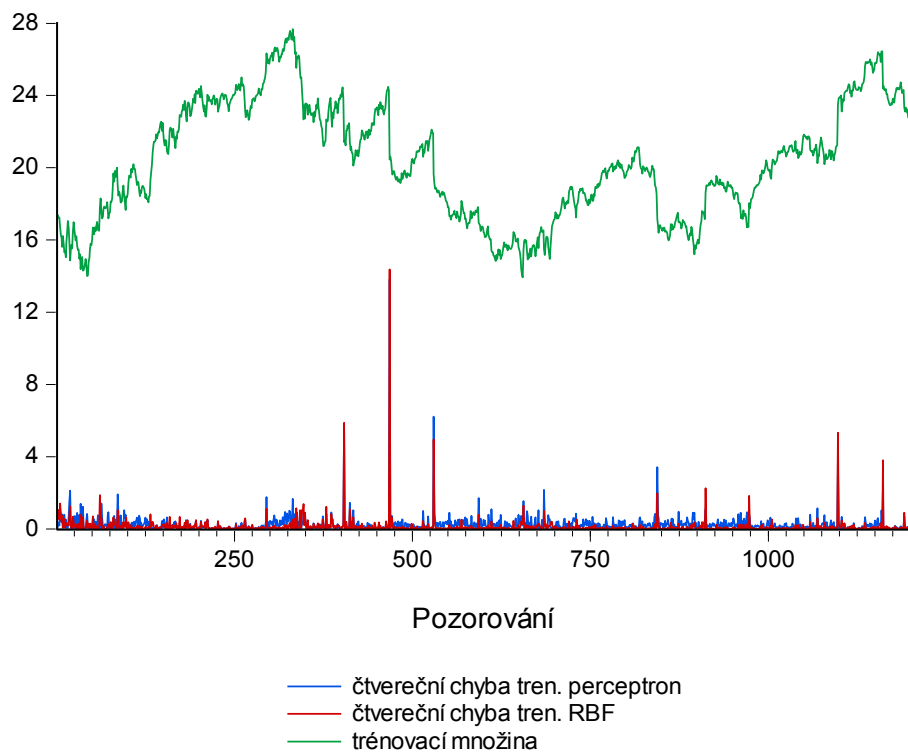


Obrázek 4.39 Srovnání trénovací množiny sítě a původní trénovací množiny

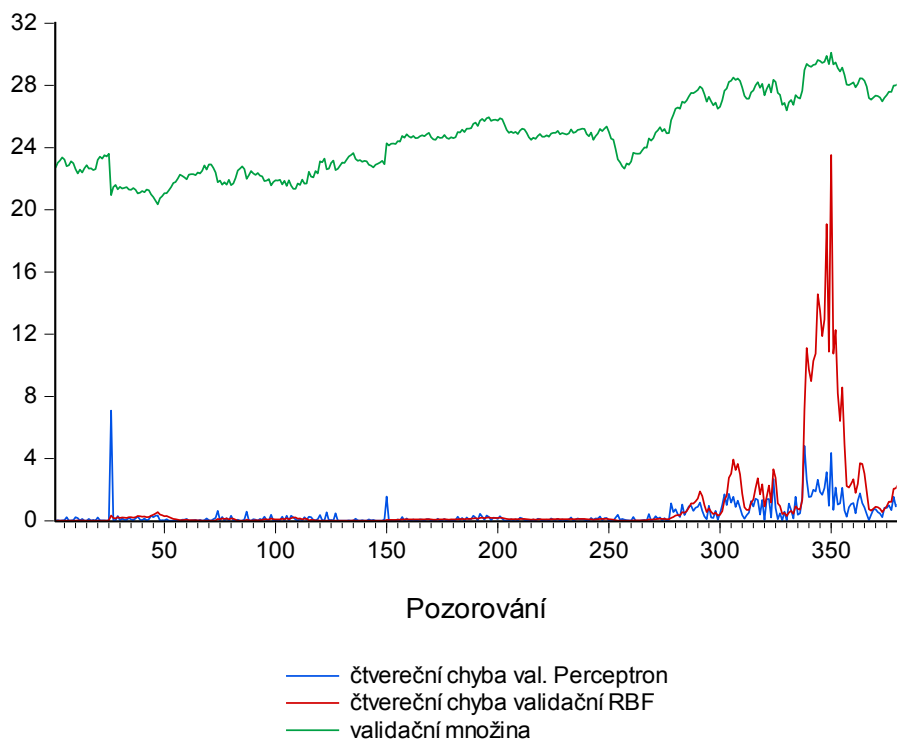


Obrázek 4.40 Srovnání validační množiny sítě a původní validační množiny

Srovnání vývoje čtvereční chyby v rámci jednotlivých pozorování neuronové sítě typu RBF a typu Perceptron, doplněné o původní validační a trénovací množinu uvádějí obrázky 4.41 a 4.42. Patrný je zde právě vývoj chyby RBF sítě na validační množině vzhledem k pozdějším pozorováním validační množiny.



Obrázek 4.41 Vývoj chyby sítí v závislosti na vývoji trénovací množiny



Obrázek 4.42 Vývoj chyby sítí v závislosti na vývoji validační množiny

4.3.3 Konečné srovnání pro časovou řadu akcií Cisco Systems

Hlavní kritérium pro srovnání predikční přesnosti nad validační množinou opět poslouží RMSE validační množiny. Přehledné srovnání jednotlivých chybových hodnot nejlepších výsledků přináší tabulka 4.9.

Tabulka 4.9 Porovnání hodnot RMSE

RMSE validační - Perceptron	0,603
RMSE validační - RBF	0,9711
RMSE validační - ARIMA	0,31483

Nejlepší hodnoty RSME validační množiny tedy vykazuje ARIMA model na hodnotě 0,31483. Oproti předcházejícím časovým řadám však v tomto případě vykazuje lepší výsledky v rámci neuronových sítí typu perceptron s hodnotou RMSE 0,603. Síť typu RBF vykazuje RMSE na hodnotě 0,9711.

4.4 Celkové hodnocení

Ve všech analyzovaných časových řadách vykazuje nejlepší výsledky statické predikce nad validační množinou ARIMA model. Neuronové sítě obou typů jsou však v těsné blízkosti a dá se říci, že vykazují velmi podobné výsledky.

Celkový problém hodnocení nástrojů v predikci je velmi komplikované. Tento fakt je dán tím, že pro statistické postupy je k dispozici úplná teorie a heuristika postupu jejího modelování nad danými daty. V případě neuronových sítí však neexistuje žádná heuristika, která by jasně navedla postup ke správnému nastavení parametrů jednotlivého druhu sítě. Teoretické nastavení parametrů sítě vězí v jejich nekonečné kombinaci. V rámci zkoušení jednotlivých parametrů není možné říci, že nějaká kombinace parametrů dosáhla nejlepšího výsledků, jakého daná síť může technicky dosáhnout. Výpočetní čas jednotlivých nastavení zabírá podle délky časové řady a komplikovanosti architektury sítí dlouhý čas. Tento čas se může na klasických pracovních stanicích pohybovat i řádu hodin, či dní. V jednotlivých pokročilých statistických softwarech, které obsahují možnost chodu neuronových sítí, se také používají pouze slepé algoritmy, které neustálým opakováním chodu sítě a průběžnou změnou parametrů jakési optimum hledají. Toto řešení však znamená potřebu velké výpočetní síly, nebo časového kapitálu pro výpočet.

Dalším problémem je, kterou z architektur zvolit pro analýzu nad danou časovou řadou. Jak je vidět z předchozích kapitol, nelze uceleně říci, že pro analýzu nad ekonomickými časovými řadami (konkrétně nad časovými řadami vývoje cen akcií) se hodí pouze jedna architektura. Lze spíše sledovat, že pro každou architekturu se hodí jiný typ dat z hlediska počtu pozorování, její volatility apod. Ačkoli byly v každém případě k dispozici až čtyři vstupní neurony, nejlepší výsledky vykazovaly sítě právě s jedním vstupem.

Jako dobře navržený a naprogramovaný se ukázal vytvořený simulátor neuronových sítí pro tuto diplomovou práci. Architektury se dají velmi dobře a snadno měnit a experimentovat s nimi. Také kompletní výpis protokolu se ukazuje jako správná cesta pro následné možnosti analýzy výsledků. V rámci potřeb je totiž možné sledovat vývoj i těch nejjemnějších kroků v rámci výpočtu neuronové sítě.

5 Závěr

Cílem této diplomové práce bylo navržení a vytvoření simulátoru chodu neuronových sítí pro vybrané architektury neuronových sítí a jeho využití při následné statické predikci ekonomických časových řad, v porovnání se statickou predikcí vykonanou současnými vybranými statistickými modely.

Teoreticko-metodologická část práce se snažila osvětlit teorii a nutné postupy k problematice statistických a soft computingových metod. Nedílnou součástí je tedy popis ekonometrických modelů, včetně etap ekonometrického modelování a podrobný popis Box-Jenkinsovy analýzy dat. V rámci soft computingových metod, tedy zejména použitých neuronových sítí, je vysvětlena ucelená struktura této problematiky. A to včetně nastínění historického vývoje, základních principů a pokročilých architektur. Mezi využitě architektury neuronových sítí v této diplomové práci patří architektura neuronové sítě typu Perceptron a neuronové sítě typu RBF.

Následující část práce se věnuje samotné problematice vytvoření vlastního simulátoru neuronových sítí typu Perceptron a typu RBF. První část této kapitoly se zabývá problematikou načítání vstupních dat a přesně popisuje její řešení a ucelený postup pro čištění a nastavení těchto dat tak, aby byly schopné vstupovat do simulátoru neuronových sítí. Dále se kapitola zabývá samostatnou tvorbou simulátoru. Samotné architektury obou typů sítí od sebe byly programově odděleny a jsou tedy dostupné skrze dva oddělené programové projekty. U jednotlivých architektur jsou představeny základní parametry a metody tříd, které jejich chody programově reprezentují. Na konec této kapitoly je uvedena problematika výstupu simulátoru. Výstupem simulátoru je pak ucelený textový protokol kompletního chodu neuronových sítí, který je programovatelně variabilní a je tak schopen uvádět pohled od samotných odhadů časových řad až po největší detaily výpočtu neuronových sítí.

Čtvrtá kapitola se zabývá ověřováním a porovnáním predikční přesnosti statistických modelů a neuronových sítí. Toto ověření je založeno na analýze tří různých druhů ekonomických časových řad vývoje cen akcií. Každá tato časová řada tak prochází prvotní analýzou a je následně postoupena ekonometrickým modelům. Konkrétně ARIMA modelům. Tyto modely jsou následně využity k predikci těchto časových řad nad určenou validační množinou. Z této predikce pak vyplývá chybový ukazatel RMSE, který je využit jako hlavní porovnávací kritérium. Časová řada pak podstupuje také modelování simulátorem neuronových sítí. Výstup simulátoru je pak graficky a číselně reprezentován jak hodnotami nad trénovací,

tak nad validační množinou. Srovnáním prochází také vývoje čtverečních chyb neuronových sítí, vzhledem k vývoji původní časové řady.

Cíl práce byl tedy splněn. Byl navržen a vytvořen provozuschopný simulátor s robustním výstupem obou architektur neuronových sítí. Výstupy tohoto simulátoru v konkrétní podobě statické predikce nad ekonomickými časovými řadami byly porovnány s výsledky statické predikce statistických modelů.

Seznam použité literatury

Literatura

ARLT, Josef, Markéta ARLTOVÁ a Eva RUBLÍKOVÁ. *Analýza ekonomických časových řad s příklady*. Praha: Vysoká škola ekonomická v Praze, 2002. 147 s. ISBN 8024503077.

CIPRA, Tomáš. *Finanční ekonometrie*. Praha: Ekopress, 2008. 538 s. ISBN 978-80-86929-43-9.

KECMAN, Vojislav. *Learning and soft computing*. Massachusetts: MIT Press, 2001. 541 p. ISBN 0262112558.

MARČEK, Dušan. *Pravdepodobnostné modelovanie a soft computing v ekonomike*. Ostrava: VŠB-TU Ostrava, 2013. 281 s. ISBN 978-80-248-2955-5.

MARČEK, Dušan. a Milan MARČEK. *Neurónové siete a ich aplikácie*. Žilina: EDIS, 2006. 223 s. ISBN 80-8070-497-X.

MARČEK, D., M. MARČEK a L. PANČÍKOVÁ. *Ekonometria a soft computing*. Žilina: EDIS, 2008. 271 s. ISBN 978-80-8070-746-0.

ŠÍMA, Jiří a Roman NERUDA. *Teoretické otázky neuronových sítí*. Praha: Matfyzpress, 1996. 390 s. ISBN 80-85863-18-9.

VONDRÁK, Ivo. *Umělá inteligence a neuronové sítě*. 2. vyd. Ostrava: VŠB - Technická univerzita Ostrava, 2002, 139 s. ISBN 80-7078-949-2.

Internetové zdroje

HANČLOVÁ, Jana a Lubor TVRDÝ. *Úvod do analýzy časových řad* [online]. Ostrava, 2013 [cit. 2015-04-20]. Dostupné z: http://gis.vsb.cz/pan-old/Skoleni_Texty/TextySkoleni/AnalyzaCasRad.pdf

KAČENKA, Petr. *Neuronové sítě*. [online]. 1998 [cit. 2015-04-20]. Dostupné z: <http://mks.mff.cuni.cz/library/NeuronoveSitePK/NeuronoveSitePK.pdf>

STATSOFT. *Úvod do neuronových sítí*. Dostupné z: http://www.statsoft.cz/file1/PDF/newsletter/2013_02_05_StatSoft_Neuronove_site_linky.pdf

Java Excel API. *A Java API to read, write, and modify Excel spreadsheets* [online]. [cit. 2015-04-20]. Dostupné z: <http://jexcelapi.sourceforge.net/>

Seznam zkratek

AIC – Akaike Information Criterion

ACF – Auto Correlation Function

AR – Auto Regressive

ARIMA – Auto Regressive Integrated Moving Average

ARMA – Auto Regressive Moving Average

B-J – Box-Jennkins

Co. – Company

Inc. – Incorporation

MA – Moving Avarage

MAPE – Mean Absolute Percentage Error

MSE – Mean Square Error

PACF – Partial Auto Correlation Function

RBF – Radial Basic Function

RMSE – Root Mean Square Error

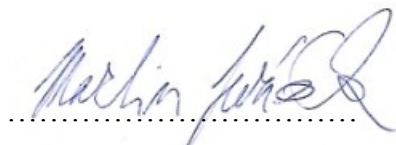
SC – Soft Computing

Prohlášení o využití výsledků diplomové práce

Prohlašuji, že

- jsem byl seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo;
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně, ke své vnitřní potřebě, diplomovou práci užít (§ 35 odst. 3);
- souhlasím s tím, že diplomová práce bude v elektronické podobě archivována v Ústřední knihovně VŠB-TUO a jeden výtisk bude uložen u vedoucího diplomové práce. Souhlasím s tím, že bibliografické údaje o diplomové práci budou zveřejněny v informačním systému VŠB-TUO;
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona;
- bylo sjednáno, že užít své dílo, diplomovou práci, nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Ostravě dne 25. dubna 2015



.....

Martin Jurásek

Seznam příloh

Příloha č. 1 – Přiložené elektronické médium

Příloha č. 1 – přiložené elektronické médium

Součástí přiloženého elektronického média je následující:

- Projekt simulátoru neuronové sítě typu Perceptron
- Projekt simulátoru neuronové sítě typu RBF
- Vstupní soubory
- Doplnkové soubory používané k analýzám
- Ukázka kompletního protokolu výstupu simulátoru